NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EMPLOYING A SECURE VIRTUAL PRIVATE NETWORK (VPN) INFRASTRUCTURE AS A GLOBAL COMMAND AND CONTROL GATEWAY TO DYNAMICALLY CONNECT AND DISCONNECT DIVERSE FORCES ON A TASK-FORCE-BY-TASK-FORCE BASIS

by

Patrick N. Kilcrease

September 2009

Thesis Advisor: Albert Barreto
Second Reader: Ross Mayfield

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 2009	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Employing a Secure Virtual Private Network (VPN) Infrastructure as a Global Command and Control Gateway to Dynamically Connect and Disconnect Diverse Forces on a Task-Force-by-Task-Force Basis	5. FUNDING NUMBERS
6. AUTHOR(S) Patrick N. Kilcrease	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

GHOSTNet is a secure and anonymous Virtual Private Network (VPN) service. Coupling Ethernet tunneling and proxy services to provide users safe and anonymous Internet access, GHOSTNet utilizes TLS (SSL) protocol with AES-256 encryption to secure the network along with PKI certificates and HMAC protection from replay attacks and UDP flooding.

This thesis will be a system level test and evaluation of the GHOSTNet infrastructure. The primary objective is to determine the functional performance of GHOSTNet as a global command and control gateway with the goal of being able to dynamically connect and disconnect diverse forces on a task-force-by-task-force basis. To accomplish this objective, a robust test and evaluation plan will be implemented to base line the system in the moderate operating conditions of COASTS field experiments conducted at Camp Roberts. The system will then be tested in various operation environments to include, but not limited to, Fort Ord, the U.S. Cost Guard Station Monterey Bay, and Southern Thailand as part of the COASTS field experimentation program spanning FY 2008 and FY2009.

14. SUBJECT TERMS Virtual Private Network, GHOSTNet, Maritime Interdiction Operations, Internet Protocol Security, Encapsulating Security Protocol, Data Encryption Standard	15. NUMBER OF PAGES 103
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

EMPLOYING A SECURE VIRTUAL PRIVATE NETWORK (VPN) INFRASTRUCTURE AS A GLOBAL COMMAND AND CONTROL GATEWAY TO DYNAMICALLY CONNECT AND DISCONNECT DIVERSE FORCES ON A TASK-FORCE-BY-TASK-FORCE BASIS

Patrick N. Kilcrease
Lieutenant, United States Navy
B.S., Norfolk State University, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2009

Author: Patrick N. Kilcrease

Approved by: Albert Barreto
Thesis Advisor

Ross Mayfield
Second Reader

Dr. Dan C. Boger
Chairman, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

GHOSTNet is a secure and anonymous Virtual Private Network (VPN) service. Coupling Ethernet tunneling and proxy services to provide users safe and anonymous Internet access, GHOSTNet utilizes TLS (SSL) protocol with AES-256 encryption to secure the network along with PKI certificates and HMAC protection from replay attacks and UDP flooding.

This thesis will be a system level test and evaluation of the GHOSTNet infrastructure. The primary objective is to determine the functional performance of GHOSTNet as a global command and control gateway with the goal of being able to dynamically connect and disconnect diverse forces on a task-force-by-task-force basis. To accomplish this objective, a robust test and evaluation plan will be implemented to base line the system in the moderate operating conditions of COASTS field experiments conducted at Camp Roberts. The system will then be tested in various operation environments to include, but not limited to, Fort Ord, the U.S. Cost Guard Station Monterey Bay, and Southern Thailand as part of the COASTS field experimentation program spanning FY 2008 and FY2009.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AES	Advanced Encryption Standard
AH	Authentication Header
C^2	Command and Control
C4ISR	Command, Control, Computers, Communications, Intelligence, and Reconnaissance
CA	Certificate Authority
CNO	Chief of Naval Operations
COTS	Commercial Off-the-Shelf
DES	Data Encryption Standard
DISA	Defense Information Systems Agency
DNS	Domain Name Server
DoD	Department of Defense
DoN	Department of the Navy
DSS	Digital Signature Standard
ESP	Encapsulating Security Protocol
FEX	Field Experiment
GD^2	Global Data Dissemination
GHOSTNet	Global Homeland Operational Secure Tactical Network
GPS	Global Positioning System
GWOT	Global War on Terror
IP	Internet Protocol
IPSec	Internet Protocol Security

ISR	Intelligence, Surveillance, and Reconnaissance
JOCC	Joint Operations Command Center
LOS	Line of Sight
MIO	Maritime Interdiction Operations
MHQ	Maritime Head Quarter
MOC	Maritime Operation Center
NIPRNET	Non-Classified Internet Protocol Router Network
PK	Public Key
SBU	Sensitive But Unclassified
SHA-1	Secure Hash Algorithm
SIPRNET	Secret Internet Protocol Network
TAO	Tactical Action Officer
UTB	Utility Boat
VBSS	Visit Board Search and Seizure
VPN	Virtual Private Network

ACKNOWLEDGMENTS

First and foremost, I would like to think my wife, Melinda, and children, Kailey and Kyler, for their understanding and patience. Their smiles and laughter kept me going every day. A special thanks to my wife for her support. She is the strength in my life, and none of this would be possible without her friendship, love, and support.

Additionally, I would like to thank Albert Barreto, Ross Mayfield, and Ryan Hale for their guidance during this process. Their expertise and experience made this thesis relevant.

Lastly, I would like to thank LT Andrew Rivas for his friendship, long hours of troubleshooting, and continual review and assistance. I wish you and your family the very best. Your friendship has been one of the greatest gifts I have been given here.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

The speed, flexibility, agility and scalability of maritime forces provide joint or combined force commanders a range of options for responding to crises. Additionally, the integrated maritime operations, either within formal alliance structures (such as the North Atlantic Treaty Organization) or more informal arrangements (such as the Global Maritime Partnership initiative), send powerful messages to would-be aggressors that the U.S. will act with others to ensure collective security and prosperity.[1]

This overwhelming task of providing global stability to promote worldwide economic commerce has specific tenants that will rely primarily on the exchange of information over tactical computer networks. The ultimate objective is to deliver timely intelligence, surveillance, and reconnaissance (ISR) necessary to achieve situational awareness by tactical and strategic decision makers throughout the chain of command and laterally among multinational partners and regional military and law enforcement through information sharing. Virtual Private Networking (VPN) is a relatively new technology currently utilized by the Department of Navy (DoN) for transmitting sensitive data across an unsecured network. Specifically, the Navy has looked at the Non-classified Internet Protocol Routing Network (NIPRNET) and methods of communicating

1 Chief of Naval Operations, "Cooperative Strategy for 21st Century Seapower," 17 October 2007, available from www.navy.mil/maritime, (accessed 05 January 2009).

Sensitive but Unclassified (SBU) data across the public Internet medium or across unsecured networks. Additionally, the Navy will be looking to provide the separation of data for different communities of interest (CIO) within the Secured Internet Protocol Routing Network (SIPRNET).[2]

The most critical aspect of this objective is providing the security of data for SBU information exchanged between multinational and DoD assets. The network infrastructure is only as good as its ability to provide security for users and information. The rapid advancements in network components, secure communications, and mobile data devices have made possible the practical use of networks in many current military and law enforcement applications in a variety of environments.

B. VISION

Envision a command cell in Norfolk, Virginia, watching a live video feed from a boarding in support of Maritime Interdiction Operation (MIO) taking place in the Port Fifth Fleet Operating Area (AOR). Key decision makers and intelligence analysts could see and hear the live interaction between boarding team members and the vessel's personnel, biometric data from the vessel's personnel could be instantly sent from a laptop, deployed with the team onboard the vessel, to a biometric database in Virginia for documentation and comparison against known or suspected terrorists. While the comparison of biometric data was being conducted boarding team members could transmit images

2 Department of the Navy, "Naval Virtual Private Network Product Requirements," 2000, 1.

of contraband, significant documents or intelligence, and observations made during the conduct of the boarding, in real time, without ever leaving the vessel's pilot house.

C. APPLICATION

Maritime forces will be employed to build confidence and trust among nations through collective security efforts that focus on common threats and mutual interests in an open, multi-polar world. To do so will require an unprecedented level of integration among our maritime forces and enhanced cooperation with the other instruments of national power, as well as the capabilities of our international partners.[3]

GHOSTNet is a cypto-analytically secure network system that has the availability to be worldwide to connect computer equipment as though they were on a single local area network.[4] GHOSTNet establishes secure communications and anonymous Internet access between multiple remote network clients via Ethernet tunneling, providing client to client communications. GHOSTNet assumes that all communication channels are unsecure and employs end-to-end encryption to achieve operation security. The modular encryption engine allows a choice of encryption method. GHOSTNet runs on most existing hardware and is easier,

3 Chief of Naval Operations, "Cooperative Strategy for 21st Century Seapower," 17 October 2007, 6.

4 Ross Mayfield, GHOSTNet, working white paper, e-mailed to author 27 January 2009.

quicker, less expensive, and more robust when compared to systems that try to achieve security by securing communication channels.

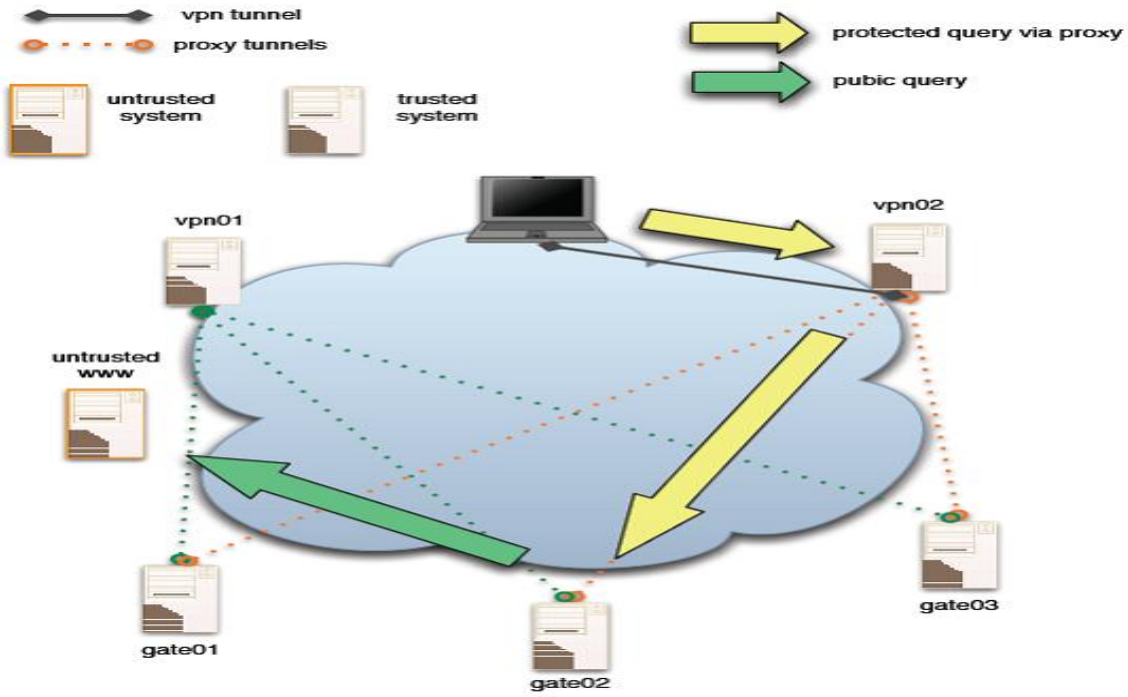

Figure 1. GHOSTNet Architecture

GHOSTNet offers a secure and anonymous Virtual Private Network (VPN) service that is unique in both implementation and features. GHOSTNet couples Ethernet tunneling and proxy services to provide users safe and anonymous Internet access. Users can connect via wired or wireless connections and on trusted and untrusted networks making GHOSTNet perfect for home users, corporate personnel, and deployed personnel.

GHOSTNet utilizes TLS (SSL) protocol with AES-256 encryption to secure the network, currently the strongest commercial encryption available. GHOSTNet also incorporated

the use of 1024 or 2048 bit PKI certificates and HMAC protection from replay attacks and UDP flooding.

This research will focus on the evaluation of the applicability and feasibility of employing a secure Virtual Private Network Infrastructure as a Global Command and Control gateway to dynamically connect and disconnect diverse forces on a task-force-by-task-force basis. Field testing areas being examined include the evaluation of video, voice, and data transmissions via a laptop computer to a remote command and control center from a maritime interdiction team conducting a MIO boarding onboard a vessel underway, or anchored. Utilizing GHOSTNet, the boarding team could securely pass all information, video, biometric data, and so forth directly to the entire MIO chain of command as it is captured. This would directly enable a MIO boarding to become safer for the VBSS teams, quicker in execution, and provide more utility in the area of intelligence gathering and documentation. From a C^2 perspective, the MIO commander, ship CO, and all respective Tactical Action Officers (TAOs) and watch captains would have instant access to the conduct and progress of the boarding in near real time. Intelligence specialists would be able to receive key items of interest as they are discovered onboard instead of hours after the vessel had been released and the boarding secured.

THIS PAGE INTENTIONALLY LEFT BLANK

II. TECHNOLOGY BACKGROUND

A. VIRTUAL PRIVATE NETWORKS

Virtual Private Networks (VPN) make use of the public Internet establishing a cost-effective secure network allowing companies to connect physically separated workers and remote business locations without the high cost of leasing or purchasing private circuits. While the company's Internet Service Provider (ISP) and other Internet users are completely unaware of the secure connection that has been established, the companies' workers are able to perform their duties as if they were physically in the office and have direct access to the resources they need.[5]

B. TUNNELING

Tunneling is the transfer of data between two similar or dissimilar networks via an intermediate network. Tunneling encloses one type of data packet into the packet of another protocol. Before the encapsulation takes place, the packets are encrypted so that the data is unreadable to anyone monitoring the network. These encapsulated packets travel through the Internet, which serves as one example of an intermediate network until they reach their destination. Upon arrival, the packets are decrypted and returned to their original format. The protocol of encapsulating packets is understood by the network and by both the points where the packet enters and exits the network. Tunneling

5 John Mairs, *VPNs; A Beginners' Guide* (Berkley, CA: McGraw Hill, 2002), 78.

enables you to place the packet that uses a protocol not used by the Internet inside an IP packet and send it securely over the Internet. You can use private, non-routable, IP addresses inside a packet that uses an assigned public, routable IP address to tunnel your private network through the Internet.

C. TUNNELING PROTOCOLS

The General Routing Encapsulation (GRE) provides a standard for tunneling data and is defined in the Institute of Electrical and Electronics Engineers (IEEE) Request for Comments (RFCs) 1701 and 1702 (available from www.ietf.org/rfc/rfc1701 and www.ietf.org/rfc/rfc1702). The concept of GRE is that a protocol header and delivery header are added to the original packet and its payload is encapsulated in the new packet. There are three main Layer 2 VPN technologies defined in the RFCs that use encryption methods and provide for user authentication: Layer 2 is the data link layer of the Open Systems Interconnection (OSI) model, a 7 layer model that defines standards of communication on routable networks.

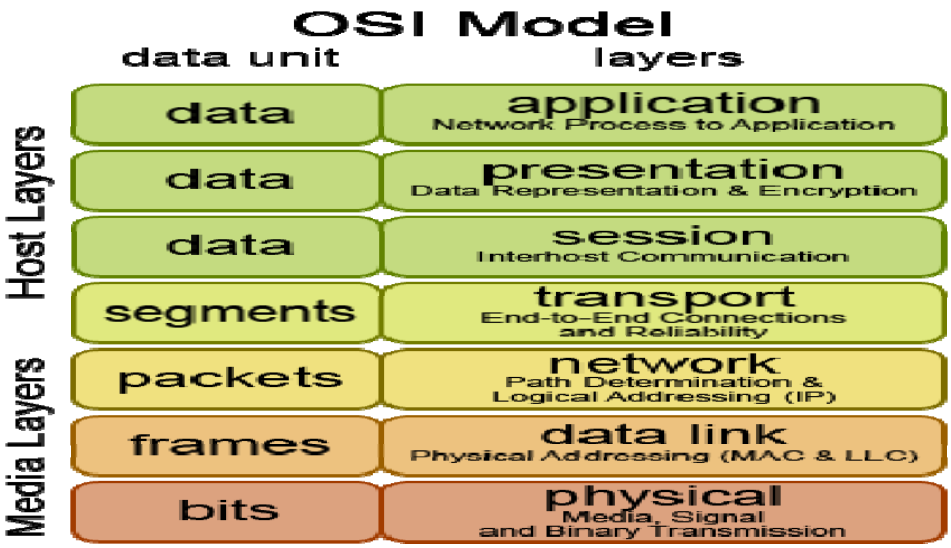

Figure 2. 7 Layer OSI Model

1. Point to Point Tunneling Protocol (PPTP). PPTP uses GRE for encapsulation and can tunnel IP, IPX, and other packages over the Internet. The main disadvantage is the restriction that there can only be one tunnel at a time between communication partners.

2. Layer 2 Tunneling Protocol (L2TP). L2TP is the industry standard. It combines he advantages of L2F and PPTP without suffering form their disadvantages. It does not provide its own security, but it can be combined with other technologies that do offer encryption such as IPSEC.

3. Layer 2 Security Protocol (L2Sec). L2SEC was developed to provide a solution to the security flaws of IPSec. The overhead, which is defined as the amount of format information stored in the packet header, that when is combined with the assembly and disassembly of packets reduces transmission speed, is huge and L2SEC uses Secure Socket Layer/Transport Layer Security (SSL/TLS).

IPSec is the most widespread tunneling technology. It was developed as the Internet Security Standard on Layer 3 by the Internet Engineering Task Force (IETF) in 1995. IPSec can be used to encapsulate any traffic of application layers, but no traffic of lower network layers. Neither, network frames, IPX, or broadcast messages can be transferred. IPSec uses a variety of encryption protocols and authentication protocols. The two prevalent methods used by IPsec are:

1. Tunnel Mode. The tunnel mode encapsulates the whole IP packet into a new packet and sends the new packet to the endpoint. This protects the addresses of the sender and recipient, as well as all other metadata, or meta-information. Metadata is defined as data about other data and can represent a datum or a collection of data.

2. Transport Mode. In transport mode, only the payload of the data is encrypted and encapsulated. This significantly reduces the overhead, but an attacker can easily read the metadata and find out who is communicating, although the data is encrypted and protected.

D. VPN SECURITY

The goals of VPN security is to: to provide privacy of data transferred, ensure the integrity of the data, and ensure that the data is available when it is needed. This is accomplished using either symmetric encryption or asymmetric encryption.

1. Symmetric Encryption. In this method, both the sender and receiver use the same key to encrypt and decrypt the message. Everyone who has the key can decrypt the

traffic, and if the key is compromised, the entire VPN is compromised. Symmetric Encryption schemes are susceptible to a multitude of attacks such as brute force attack (a method for breaking cryptographic systems by systematically trying a large number of keys in a key space in order to decrypt a message), and man-in-the-middle attacks (attacker intercepting the data between sender and receiver, copy, and forward without the sender or receiver realizing their traffic has been intercepted). Systems utilizing a symmetric encryption scheme should change keys frequently and utilize a combination of key lifetime and key length to ensure that an attacker cannot decrypt the key while it is valid.

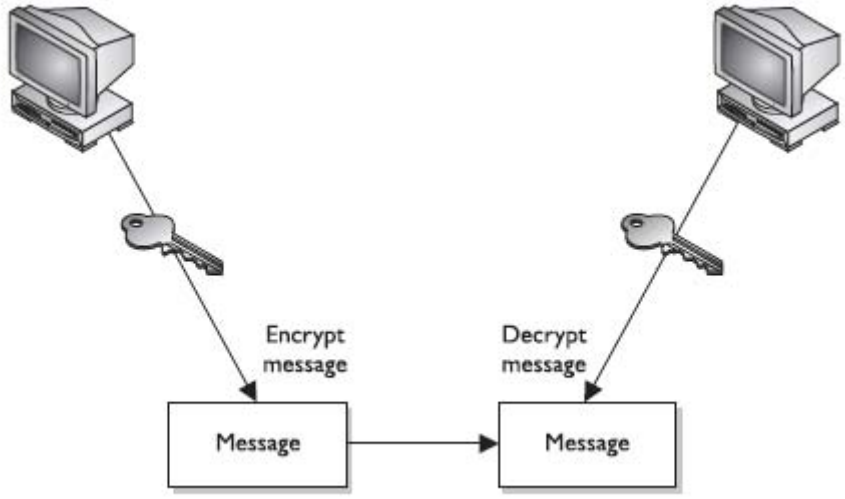

Figure 3. Symmetric Encryption System (All In One CISSP)[6]

6 Shon Harris, *All In One CISSP Exam Guide* (San Francisco, CA: McGraw Hill, 2008), 680.

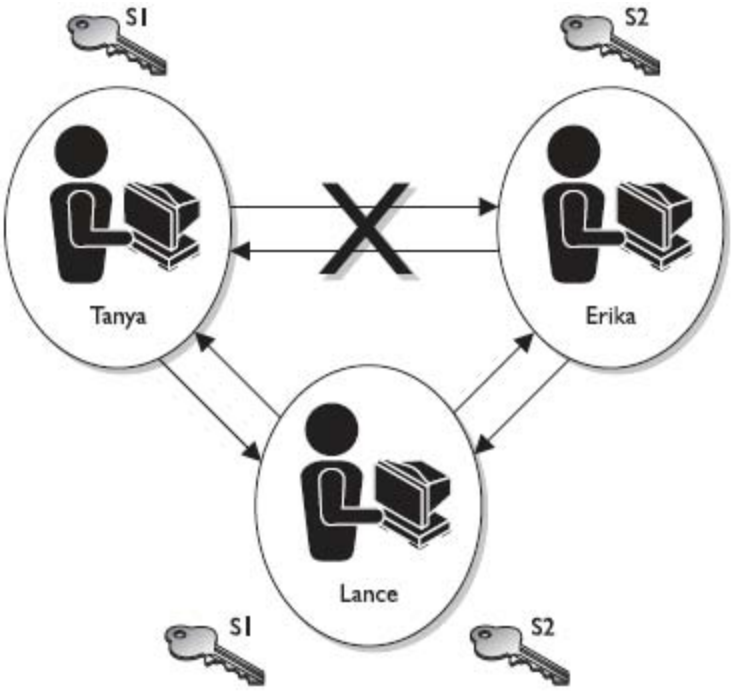

Figure 4. Man in the Middle Attack (All In One CISSP)[7]

2. Asymmetric Encryption. In asymmetric cryptography, each user has a public and a private key. The public key is known to everyone, and it is used to encrypt the message, and since both keys are created linked by a mathematical algorithm, only the receivers private key can decrypt the messaged. In this public key system, users must ensure that their private key is kept secure.

7 Shon Harris, *All In One CISSP Exam Guide*, 681.

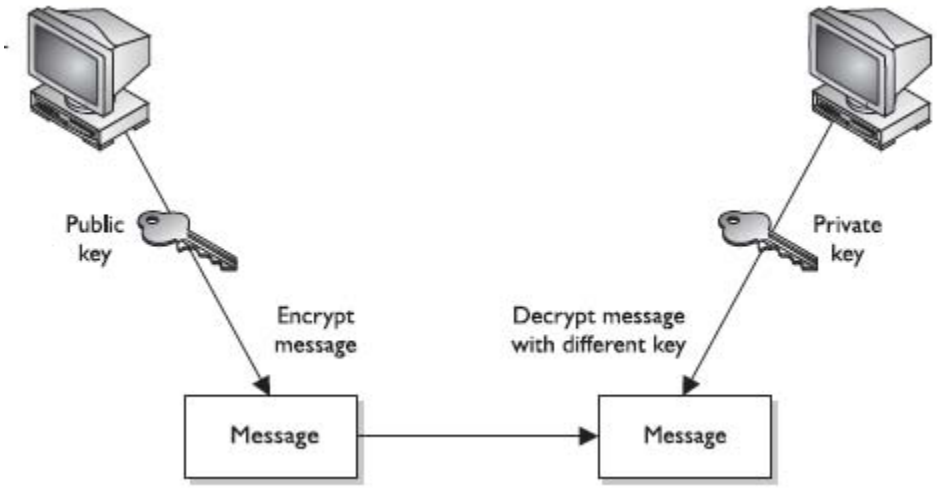

Figure 5. Asymmetric Encryption System (All In One CISSP)[8]

3. Hash Function. A hash function takes a block of data and returns a fixed-size bit string, hash value, such that an accidental or intentional change to the data will change the hash value. The ideal hash function has four main properties: it is easy to compute the hash for any given data, it is extremely difficult to construct a test that has a given hash, it is extremely difficult to modify a given test without changing its hash, and it is extremely unlikely that two different messages will have the same hash. Hash functions are used for message integrity checks, ensuring the data is whole or complete, digital signatures, a form of asymmetric cryptography to ensure non-repudiation, authentication, and other various information security applications.

8 Shon Harris. *All In One CISSP Exam Guide,* 701.

E. DIFFIE-HELLMAN KEY EXCHANGE

The Diffe-Hellman encryption system was developed in 1976 to solve the problem of key distribution for private key encryption systems and the need to find a secure way of deciding on a private key using the same method of communications that you are trying to protect.[9] This enables users to exchange symmetric keys making prior arrangements. The Diffie-Hellman algorithm is susceptible to the man-in the-middle attack (see Figure 4) and thus other protocol, such as authentication, prior to exchanging public keys.

9 Eric Maiwald, *Fundamentals of Network Security* (Burr Ridge, IL: McGraw-Hill, 2004), 324.

III. GHOSTNET SETUP

A. INSTALLATION OF OPENVPN FOR THE SERVER AND CLIENT

OpenVPN software can be downloaded from www.openvpn.net/downloads. OpenVPN runs on most operating systems (OS) to include Microsoft© Windows 2000/XP/Vista, Solaris, BSD, and Mac OS X. The installation of OpenVPN is standard for both the client and the server. Once the configuration is completed and tested, the client configuration file can be copied and used for all other clients, just ensure that you change the name of the client's key and cert to match the ones provided for that client.

1. End User Installation

GHOSTNet uses an open source set up file, OpenVPN, for installation. The systems must be able to support the Universal TUN/TAP drivers. The easiest installation would be to install the OpenVPN GUI that will allow the opening and closing of tunnels by the user. GHOSTNet can also be run as a service and will start automatically on startup. Once you have downloaded the installation files, set up is accomplished by following the setup wizard.

a. Accept the end user license agreement.

b. Select the components and services you want to install (the standard components make sense in most applications).

c. Select an installation directory (in most cases this will be c:\program files\openvpn).

d. The wizard will complete the installation.

After the installation is finished, you must copy the correct configuration file along with keys that are provided by your network administrator into the c:\program files\openvpn directory.

2. Establishing Connection with a GHOSTNet Secure Server

To establish the secure connection with the GHOSTNet server, ensure you are connected to a network via an Ethernet cable or wirelessly, and right click on the shortcut in the toolbar and select the GHOSTNet_1194 connection. This will run the configuration file located in the c:\program files\openVPN directory. Once the connection is established, you will see the green computers indicating a secure connection in the system icon tray.

3. Verifying the Secure Connection

Once the connection has been established and the OpenVPN GUI in the windows toolbar displays a secure connection (changes color from red to green), you can verify you have a secure connection by accomplishing the following steps:

To verify that your network traffic is being routed through the tunnel created, use a web browser to go to www.whatismyipaddress.com. This will indicate that you have an IP address from Road Proxy and you are being routed from Greensboro, North Carolina, or New Haven, Connecticut.

Browse to www.xxx.com to check your proxy connection. A proxy is an intermediary between an external and internal

application. This intermediary usually pretends to be the end point for both sides of the connection and accepts the client request, rewrites it and sends it to the server. Return traffic is handled the same way. Application Level Gateway (ALG) and Web Proxy are common name for devices that do this method of mediating traffic is slower than normal as the gateway must decode/encode the packets and extra time.

B. RUNNING OPENVPN AS A SERVER ON WINDOWS

To run OpenVPN as a server in Microsoft Windows©, complete the install as previously described for the client. Then go to the control panel, administrative tools, services, and service manager. Find the entry OpenVPN Service and double click the entry. Under startup, select automatic. OpenVPN will know try to start a tunnel for every .ovpn file it finds in the config directory.[10] A sample configuration file for the server can be located in Appendix B. Ensure that the router has been set up for port forwarding and that an entry exist in the advanced routing to properly route the VPN traffic to the server.

10 Markus Feilner, *OpenVPN: Building and Integrating Virtual Private Networks* (Brimingham, UK: Packt Publishing Ltd., 2006), 95.

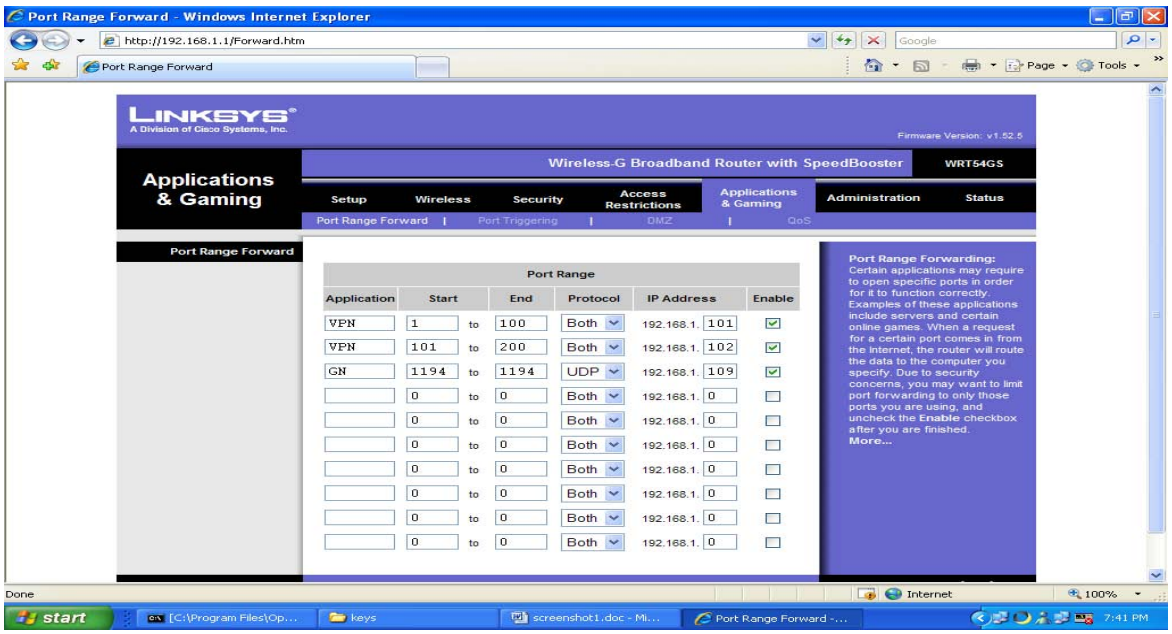

Figure 6. Linksys® WRTG54GS Port Range Forwarding entry

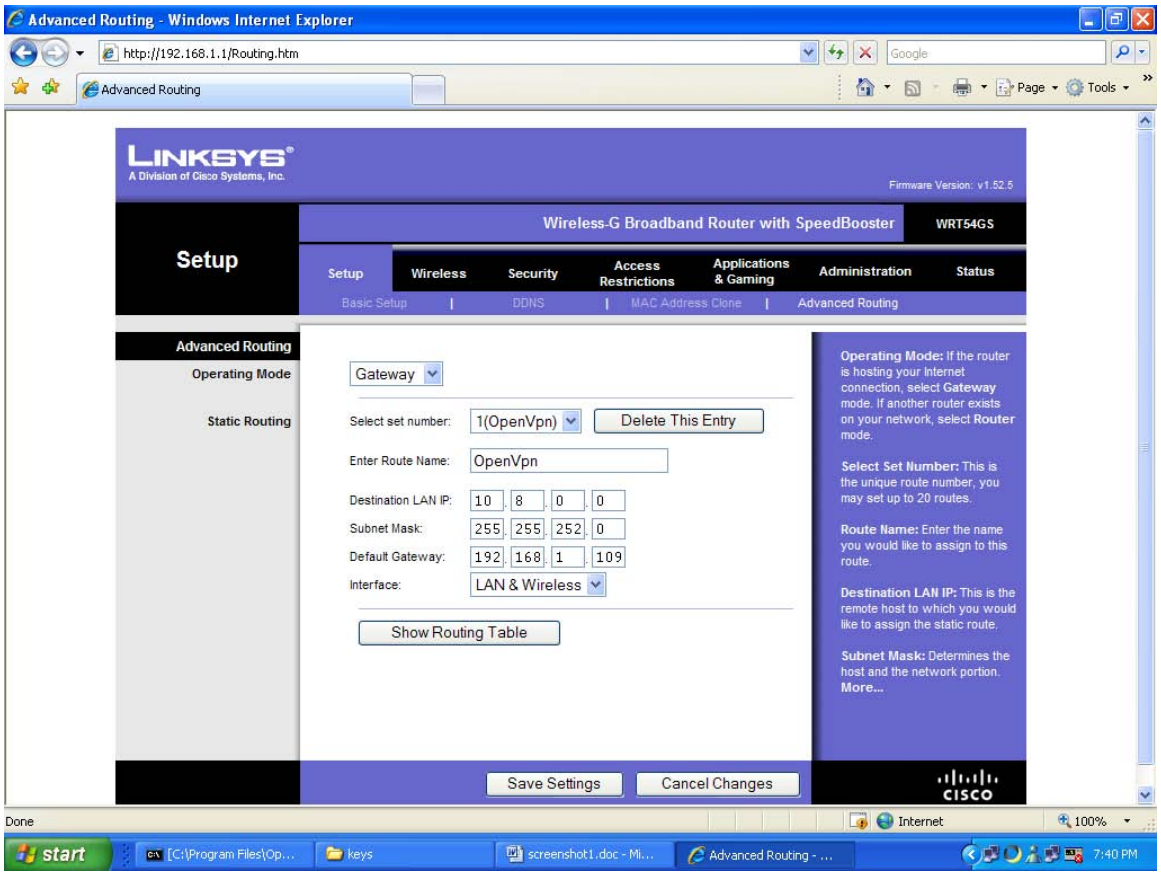

Figure 7. Linksys® WRTG65GS Advanced Routing entry

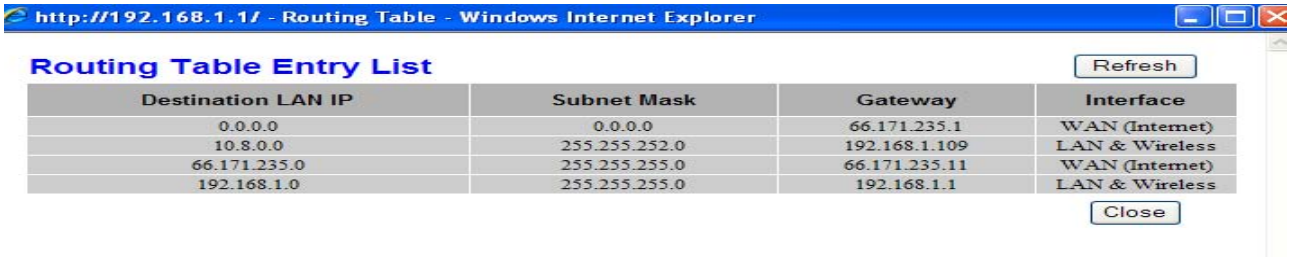

Figure 8. Linksys® WRT54GS Routing Table entry

1. Creating x509 Certificates

In the OpenVPN\easy-rsa directory, create a folder named keys and copy the files serial.start and index.txt.start into the folder. Rename the files to serial and index.txt respectively.[11] These files will be used as a database for certificate generation.

To prepare the standard configuration for the certificates, double click on the c:\program files\openvpn\easy-rsa\init-config.bat file to copy a template of the vars.bat.sample to vars.bat and openssl.cnf.sample to openvpn.ssl. Next, you will need to edit the vars.bat file (this file contains variables used by OpenVPN's scripts to create certificates).[12] Right click on the vars.bat file. The changes you make in the vars.bat file standardize the key generation process and prevent you from having to continually enter in the data.

11 Markus Feilner, *OpenVPN: Building and Integrating Virtual Private Networks,* 110.

12 Ibid., 111.

Set HOME=%ProgramFiles%\OpenVPN\ easy-rsa	Leave as specified
setKey-Config=openssl.cnf	Leave as specified
Set Key_Dir=keys	Leave as specified
Set Key_Size=1024	Change to 2048
Set Key_Country=US	Change as needed
Set Key_Province=CA	Change as needed
Set Key_City=SanFrancisco	Change as needed
Set Key_ORG=FortFunston	Change as needed
Set Key-EMAIl=mail@hsot.domain	Change as needed

Table 1. Vars.bat entries

2. Creating the Diffie-Hellman Key

To create the keys that will be used for encryption, authentication, and key exchange start the batch file from the command prompt c:\ProgramFiles\OpenVPN\easy-rsa\build-dh.bat. The Diffie-Hellman key will be generated for you.

3. Building the Certificate Authority

From the command prompt, enter build-ca.bat. This batch file generates a self signed certificate for the CA that can be used to create and sign client certificates to authenticate other machines. If you do not need to change any of the data entered in the vars.bat file entered in para E simply press enter and the certificate for the CA will be generated in the keys directory.

4. Generating Server and Client Keys

To generate the server key and have it signed by the CA created in para G., type “build-key-server.bat “the name of your VPN Server.”” This will generate a 2048-bit private key. You will be asked for extra attributes other than

those provided from the vars.bat file, including the ability for a password. If you choose to enter a password, no one can set up a connection without this password. After the certificate is generated, you will be asked if you want to have it signed by the CA. Enter ‘Y’ twice to have it signed.

To create the client keys enter “build-key.bat “name of the client”” and follow the steps outlined above. Additionally, ensure you enter a unique “Common Name” for every client key that you create, otherwise the key will not be signed by the CA and you will not be able to authenticate with the server.

5. Keys to Transfer to the Client

Three files must be transferred to the client securely to establish the connection; vpn-client.crt, vpn-client.key, and ca.crt.

6. Configuring OpenVPN to Use Certificates

A sample configuration file for the server and client has been included in Appendix B, and during install, sample configuration files are available in the sample configuration folder. To configure OpenVPN server to use the certificates created, open the configuration file in Notepad. Ensure the following entries are made, noting the mapping to the correct folder for the certificates and keys. You will have to securely deliver the client their certificate, key, CA cert, and the dh2048.pem file.

Server	Client
tls-server	tls-client
dh keys/dh2048.pem	Dh keys/dh2048.pem
ca keys/ca.crt	Ca keys/ca.cert
cert keys/vpn-Server.crt	Cert keys/Vpn-Client.crt
key keys/vpn-server.key	Key keys/vpn-client.key

Table 2. x509 certificate configuration file entries

IV. EXPERIMENT METHODOLOGY

A. COASTS

Cooperative Operations and Applied Science & Technology Studies (COASTS) is an international field experimentation program at the Naval Postgraduate School, designed to develop and assess commercial off-the-shelf (COTS) and leading edge technologies for specific military, peacekeeping and stability operations, law enforcement, and first responder missions. COASTS engages international and domestic partners at the research and development (R&D) level through cooperative science and technology field experimentation to investigate and match participant mission needs with integrated command and control, computers, communications, intelligence, surveillance and reconnaissance (C4ISR) solutions in domestic, bi-lateral and multi-national environments.

COASTS conducts integrated, multi-phase scenarios to demonstrate and evaluate these C4ISR solutions over a series of five field experiments (FEX I – V), which ultimately culminate at FEX V in the final demonstration scenarios.

1. Technical Overview

The scope of the specific research encompasses the technology related to the applicability and feasibility of connecting users via a VPN utilizing the technologies of an emerging capability—called GHOSTNet—to remotely view streaming video, control desktop applications, participate in video teleconferencing using various wireless networks

via simulation and field experimentation. GHOSTNet can be used as the underlying infrastructure to connect any GHOSTNet enabled device to any other GHOSTNet enabled device in a mission specific manner.

The GHOSTNet open VPN application is used to establish secure communications between multiple remote network clients via tunneling. Tunneling is a process by which a client can access a Private network utilizing the Public Internet, given the proper security capabilities and configurations of both the client and the network in order to establish the connection.

2. FEX-II/III

In January and February 2008, Field Exercise II and III were conducted and utilized as local site survey evolutions and network preparation exercises to support field requirements for FEX IV and V, the COASTS-08 final scenario and demonstrations in Thailand. McMillen Airfield in Camp Roberts, CA, was the site of both FEX II and III, the second and third iterations of COASTS field exercises, which took place in the vicinity of a runway and transiting installations. This strategic location provided a chance to deploy and test realistic network topologies and link scenarios. The physical network architecture and layout that would be employed in Thailand during FEX IV and V was constructed using the runway, installations, and roads surrounding the McMillen Airstrip, see Figure 9, below. Valuable lessons learned and network requirements were gained from performing the required tests on the actual

equipment configurations that were planned to be demonstrated in the following months in Thailand.

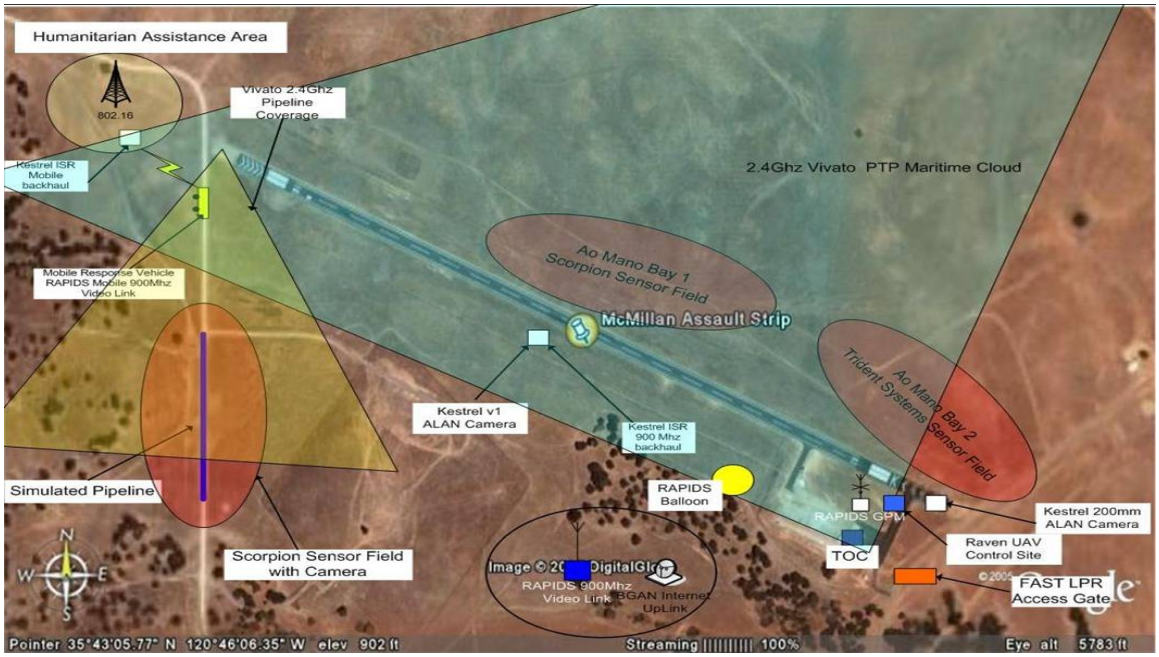

Figure 9. Network Architecture for FEX II/III

Also, contributing to FEX III was the United States Coast Guard (USCG) from their USCG Station at Monterey Bay. The USCG provided the use of a 41-foot utility boat for use in testing while underway in Monterey Bay. It was specifically an opportunity to conduct proof of concept testing for the wireless network provided by the Vivato phased array base station, the Ruckus device, and the GHOSTNet application. This testing configuration would mimic the underway demonstration that would occur later in Thailand.[13]

13 Andrew P. Rivas, "Implementation of Phased Array Antenna Technology Providing a Wireless Local Area Network to Enhance Port Security and Maritime Interdiction Operations," Master's thesis, in progress, Naval Postgraduate School.

3. FEX-IV/V

Field Exercise IV and V, conducted in March and May 2008, respectively, took place at Ao Manao Airbase in Prachuap Khiri Khan, Thailand, which is located approximately 312 km, or 194 miles, south of Bangkok. All successful implementations of equipment and experiments at FEX II and III would be deployed at Ao Manao for further operational testing and development. For the Vivato – GHOSTNet wireless network, FEX IV and V would be used to tie together network tests conducted at Monterey Bay and Camp Roberts.

4. Scope of Testing

The intent of the Vivato—GHOSTNet testing was to demonstrate the feasibility of utilizing an wireless 802.11g network over water and land in order to provide secure Global Data Dissemination (GD^2) to physically remote operational commanders and their staff. From a remote C^2 center, the planning staff would have the ability to view and receive video, voice, and data from a naval unit conducting a maritime interdiction operation (MIO) boarding on an underway vessel.

5. Selected Metrics

There were multiple protocols available to demonstrate video streaming (e.g., TCP/IP, UDP, RTSP, Unicast, Multicast and P2P), but only TCP/IP was tested since it is directly attributed to the scope of this thesis. The metrics used for these tests were: *throughput*—as measured by bulk

transport capacity; *response time*—as measured by roundtrip delay and loss; and *video streaming*—as measured by throughput thresholds on video packets.

6. Throughput

Throughput measures the maximum amount of *intended* data transferred across a communications link or network. It does not include any additional packets or encryption overhead that may be transferred due to strong encryption schemes implemented or multiple data transmissions over the wireless medium which would constitute the total data rate. The method used to perform this measurement is to transfer a “large” file (3 Mb) between two network nodes and measure the time taken to receive the file. The throughput is then calculated by dividing the file size by the time to get the intended data in megabits, kilobits, or bits per second.

7. Response Time

Response time is a measure of effectiveness related to the amount of time it takes a data packet to traverse a given distance. Essentially, it is the elapsed time between the end of an inquiry on a computer system and the beginning of a response. Network performance monitoring tools were configured to measure and display various parameters characterizing communications between or among a pair of network endpoints, or nodes. In TCP/IP-based networks, one such parameter was the network Round Trip Time (RTT).

As a control measure, the RTT was measured from the "Chariot Box," the shore-based endpoint location, to eliminate any inconsistencies related to tests taken at various locations.

8. Video Streaming

Video streaming refers to the ability of an application to play synchronized video media streams, in a continuous way, while they are being transmitted to the client over a data network

9. Measures of Effectiveness and Performance

The COASTS-08 Field Exercises provided an environment in which to test the qualitative measurements of the Vivato—GHOSTNet Network. The Measures of Performance (MOPs) directed that bandwidth and throughput performance of a network were the most important factors for testing. The qualitative measures were formed by reviewing the COASTS-06 and COAST-07 after action reports (AARs) that showed considerable network degradation during high bandwidth usage and video streaming evolutions. For the Measures of Effectiveness (MOE), Ixia's IxChariot was implemented to collect, display, and analyze the pertinent information related to network performance characteristics referenced in the selected metrics listed above. The raw test data was collected and downloaded into a Comma Separated 'CSV' and Hyper Text Markup Language 'html' for compatibility reasons, and easier analysis at a later date.

10. Test Equipment

Three laptops were used for the field testing of the Vivato—GHOSTNet Network. The laptops' specifications are as follows:

1. Dell Inspiron 5100 (Chariot Box/EndPoint1) (1.0 GHz Intel Pentium II processor; 512MB RAM; and Microsoft Windows© XP, service pack 2)

2. Two Apple® Mac Books (Endpoints 2/3). (2.4 GHz Intel Core 2 Duo processor; 2GB RAM; Mac OS X and Windows XP service pack 2 – running off of VMware)

11. Testing Software

Ixia IxChariot was used as the basic network software package to conduct all network tests. IxChariot is a software tool for simulating real-world applications to predict device and system performance under realistic load conditions by utilizing packet generation and analysis. Comprised of the IxChariot Console, Performance Endpoints, and IxProfile, the IxChariot product offers thorough network performance assessment and device testing by simulating hundreds of protocols across the network.

B. FIELD TESTING CONCEPT OF OPERATIONS

1. Proof of Concept Testing

The purpose of this preliminary phase of testing was to execute a basic test and evaluation plan under reasonably moderate operating conditions on Monterey Bay before expending time and resources to field tests in Thailand.

With the assistance of the United States Coast Guard (USCG) Station Monterey Bay, underway testing commenced on February 14, 2008.

At the USCG Station, one Vivato base station was mounted overlooking Monterey Bay, at a height of eye of approximately 25 feet, and connected to the Internet. A Dell laptop was associated to the Vivato network via Wireless 802.11g at the USCG Station and served as the shore endpoint for the data packet transfer test from the Macbook (Macintosh laptop) endpoint underway. Aboard a 41-foot utility boat (UTB), a Ruckus device [router] was mounted to the mast and an Axis 213 camera was configured to a Dell laptop and secured inside the pilot house to pass video across the wireless network. The Macbook was connected via Wireless 802.11g to the Ruckus router in order to transmit data packets across the wireless network.

The focus of the testing was to observe the ability to transfer data packets and provide streaming video from the utility boat (laptops) to the USCG Station ashore and to a remote command center (JOCC at McMillen Airfield) over 100 miles away at Camp Roberts, CA. See Figure 10.

2. Observations from Initial Testing at the USCG Station

On February 14, 2008, the day of testing, a small craft advisory was issued for the area, indicating swells of to x feet and wind between 22-33 knots.

Due to the rough conditions on the bay, maneuverability was limited. The performance limitations of the wireless network included the height of the swells observed on

Monterey Bay, which affected the line of sight (LOS) of the Ruckus device [mounted to the UTB mast] and the phased array antenna [mounted at USCG Station]. The Ruckus device ceased to operate correctly after it experienced seawater intrusion, following the 3 NM test, due to sea swells contacting the mast and the pilot house of the UTB. The Macbook maintained association with the Vivato phased array antenna throughout the underway testing. Video streaming from the IP camera was successfully observed, at 2 and 3 NM tests, from the USCG Station Dell laptop and a Dell laptop at McMillen Airfield on Camp Roberts, CA, via the GHOSTNet application.

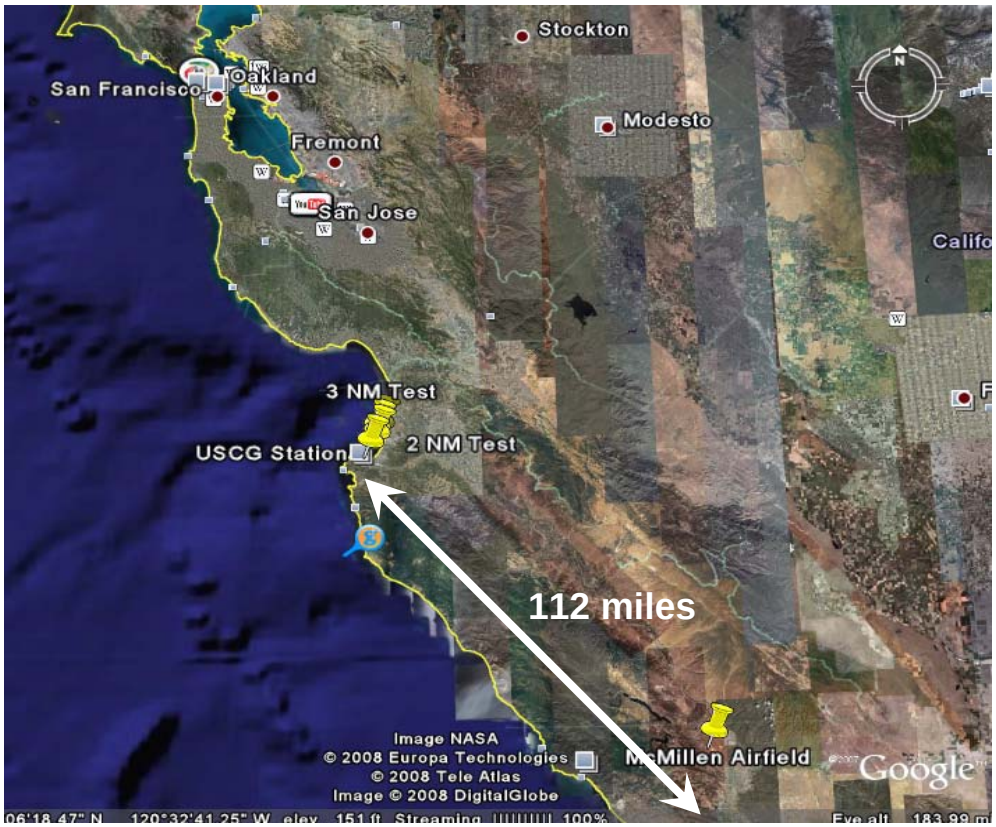

Figure 10. Distance between Monterey Bay and Camp Roberts, CA. (From: GoogleEarth)

Figure 11. Proof of Concept testing on Monterey Bay. (From: GoogleEarth)

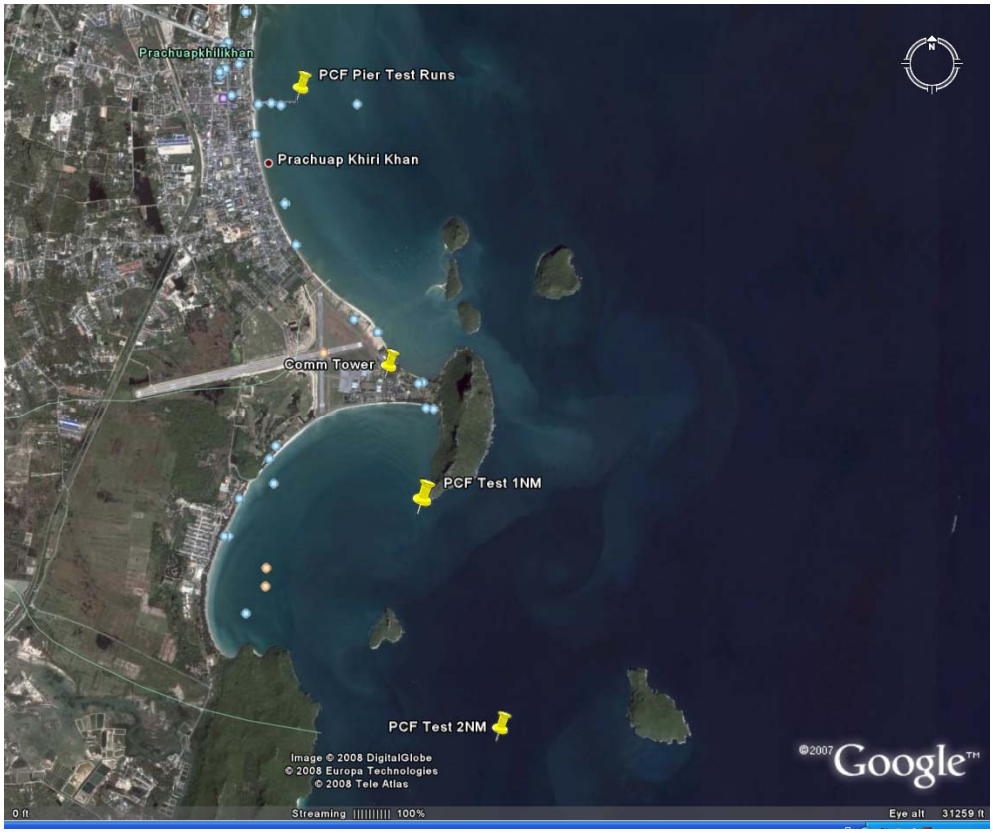

Figure 12. GPS plots of FEX-IV over-water tests. (From: GoogleEarth)

3. Observations from Ao-Mano, Thailand

Testing in Ao-Mano, Thailand, was conducted from March 24–25 2008. Both days were calm with visibility approximately 3.5–4 miles. The high temperature on 24 March was 96 degrees Fahrenheit and a low of 76 degrees. On 25 March, the high was 91 degrees with a low of 77 degrees. Both days provided a barometric pressure of approximately 29.80 inches with humidity averaging about 90%. Five tests were conducted over the two days.

a. Test One: Vivato Baseline Test

The first test was a baseline test of the Vivato network. This allowed the capturing of the response time with no users online and no encryption on the network. Due to time restrictions, only three tests were conducted with GHOSTNet enabled and six were conducted without GHOSTNet enabled.

The response time with GHOSTNet enabled measured at 1.223 seconds with a minimum time of .880 seconds, a maximum time of 19.2, and average 95% confidence interval (CI) of .7328. The average throughput measured at .01 Mbps with a minimum throughput of .001 Mbps, a maximum throughput of .012 Mbps, and a 95% CI of .0018. The average transaction rate measured at .9702 seconds with a minimum rate of .052 seconds, a maximum rate of 1.137 seconds, and a 95% CI of .1437.

The average response time without GHOSTNet enabled measured at .1533 with a minimum time of .004 seconds, a maximum of time of 18.104 seconds and a 95% CI of .2513. The average throughput measured at .2485 Mbps with a minimum throughput of .001 Mbps, a maximum throughput of 2.476 Mbps, and a 95% CI of .1307. The average transaction rate measured at 24.9467 seconds with a minimum rate of .055 seconds, a maximum rate of 227.273 seconds, and a 95% CI of 12.5855.

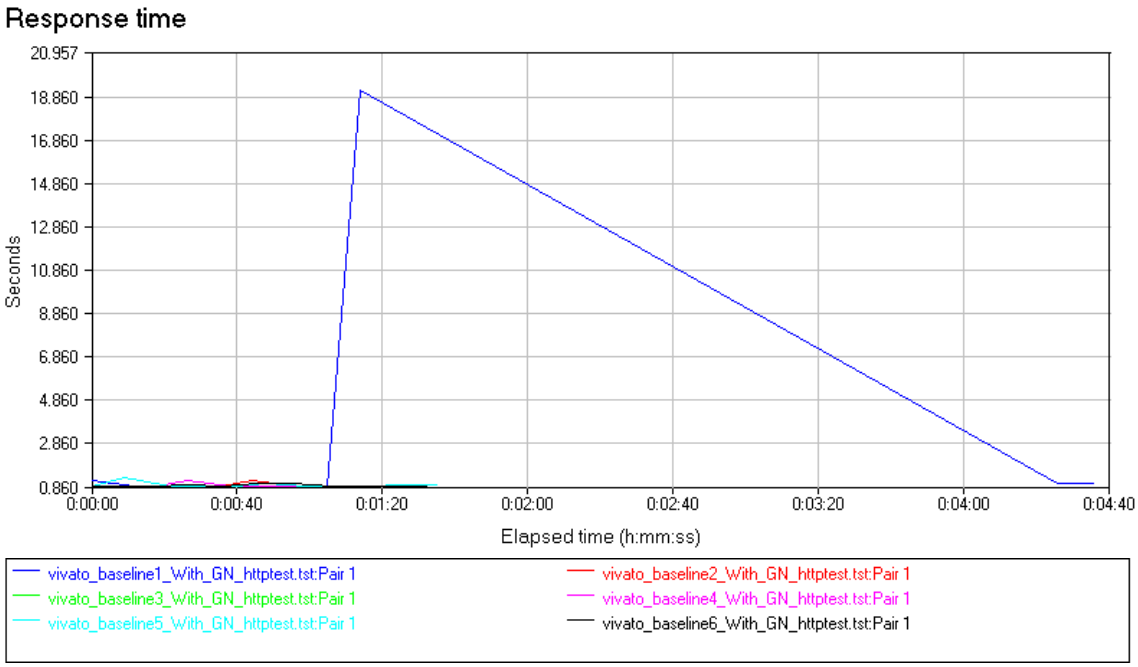

Figure 13. Vivato Response Time Baseline with GHOSTNet Enabled

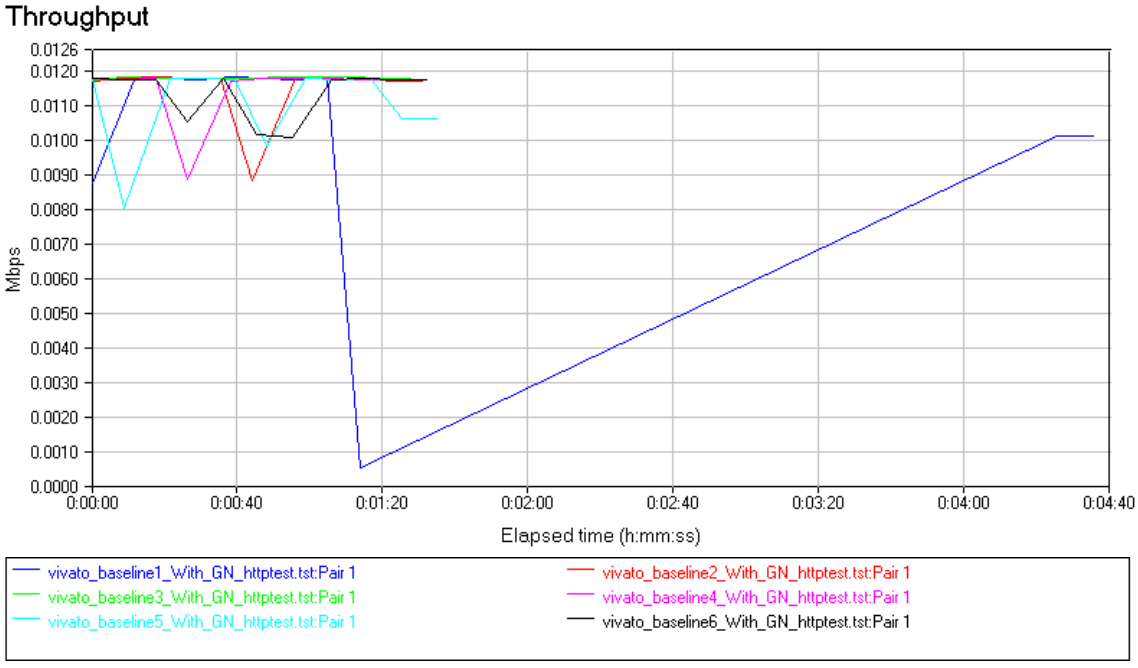

Figure 14. Vivato Throughput Baseline with GHOSTNet Enabled

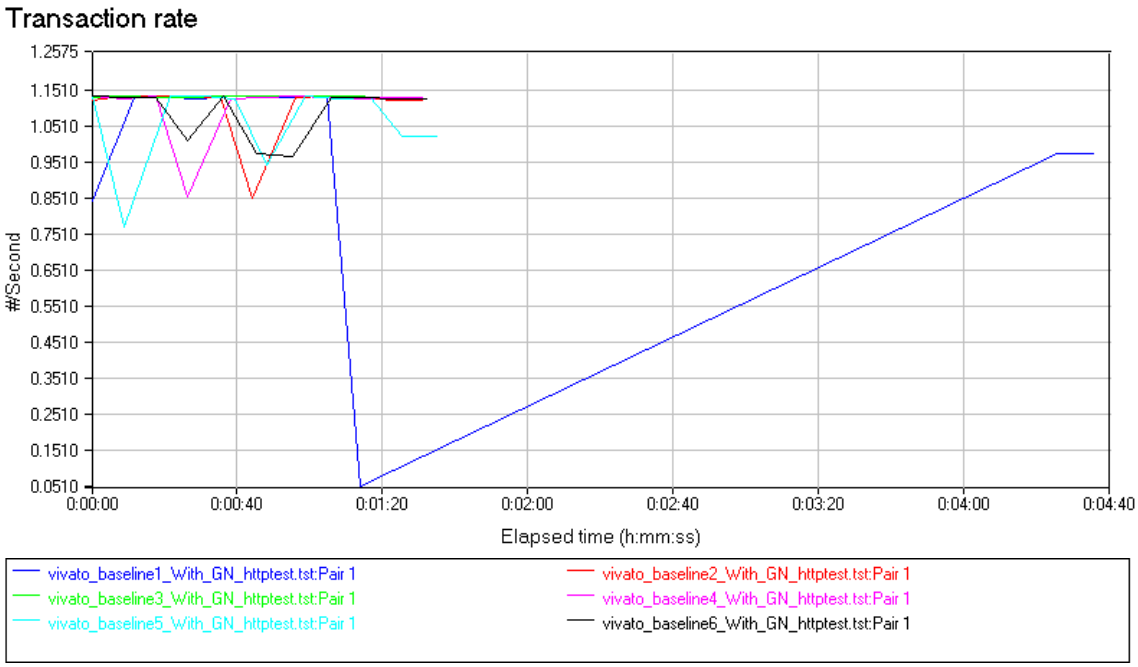

Figure 15. Vivato Transaction Rate Baseline with GHOSTNET Enabled

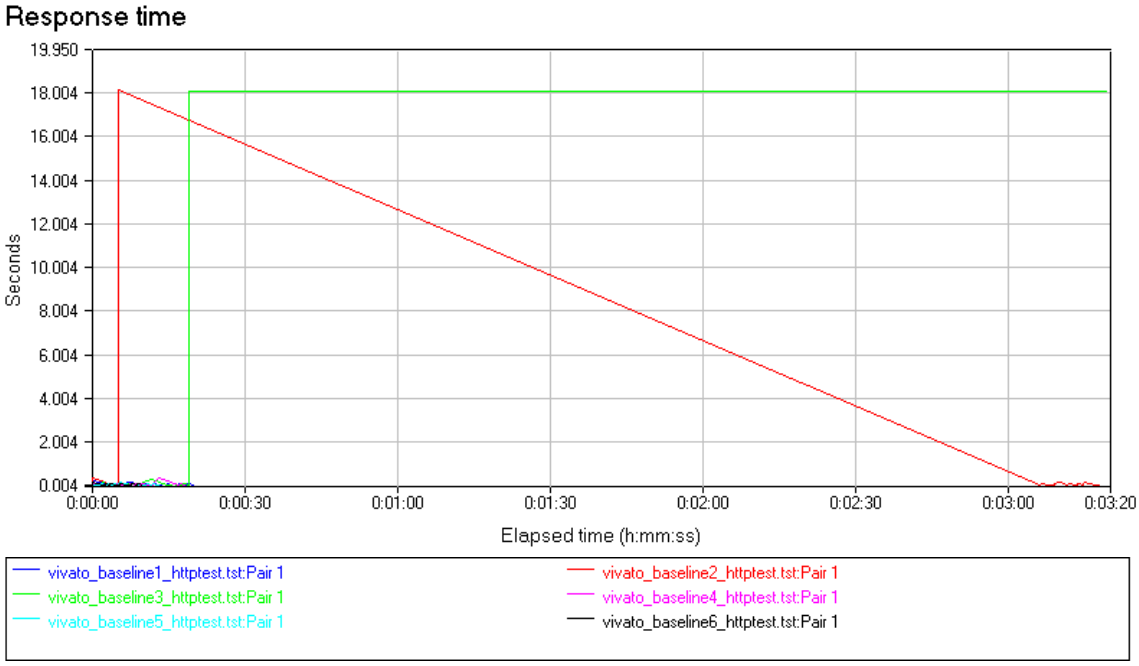

Figure 16. Vivato Response Time Baseline without GHOSTNET Enabled

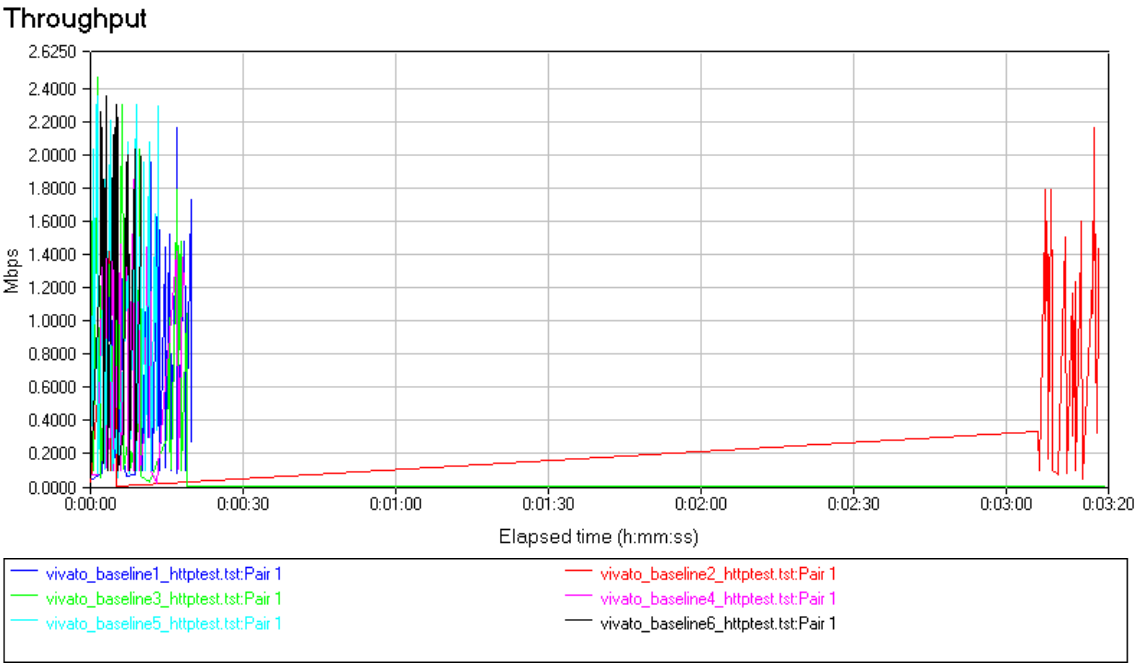

Figure 17. Vivato Throughput Baseline without GHOSTNet Enabled

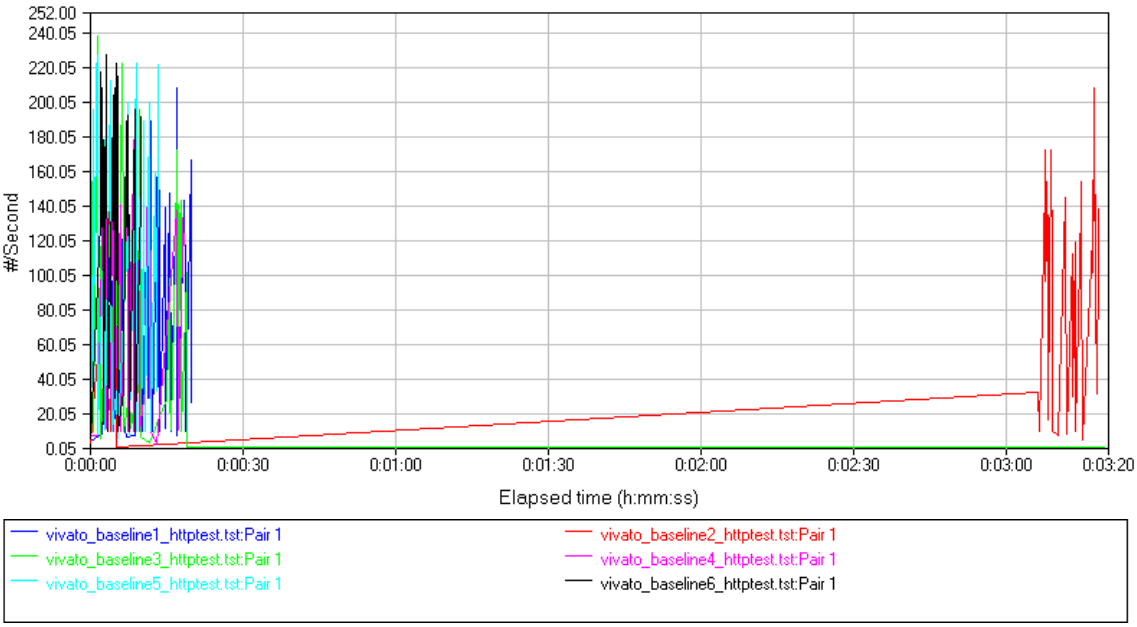

Figure 18. Vivato Transaction Rate Baseline without GHOSTNet Enabled

b. Test Two: Communications Tower to Prachuap Beach Hotel

The second test was conducted between endpoint 1 located at the communications tower (11˚ 47′ 10″ N/099˚ 48′ 34 E) and endpoint 2 located at the Prachuap Beach Hotel (11˚ 48′ 08″ N/099˚ 47′ 58″ E).

The average response time with GHOSTNET enabled measured at 1.181 seconds with a minimum time of .884 seconds, a maximum time of 3.041 seconds and a 95% CI of .4087. The average throughput measured at .009 Mbps with a minimum throughput of .003 Mbps, a maximum throughput of .012 Mbps, and a 95% CI of .0027. The average transaction rate measured was .858 seconds with a minimum rate of .329 seconds, a maximum rate of 1.144, and a 95% CI of .2643.

The average response time without GHOSTNet enabled measured at.0375 seconds with a minimum time of .005 seconds, a maximum time of .894 seconds and a 95% CI of .0368. The average throughput measured was .3260 Mbps with a minimum of throughput .012 Mbps, a maximum throughput of 2.0 Mbps, and a 95% CI of .3525. The average transaction rate measured at 31.517 seconds with a minimum rate of 1.119 seconds, a maximum rate of 192.308 seconds, and a 95% CI of 33.8985.

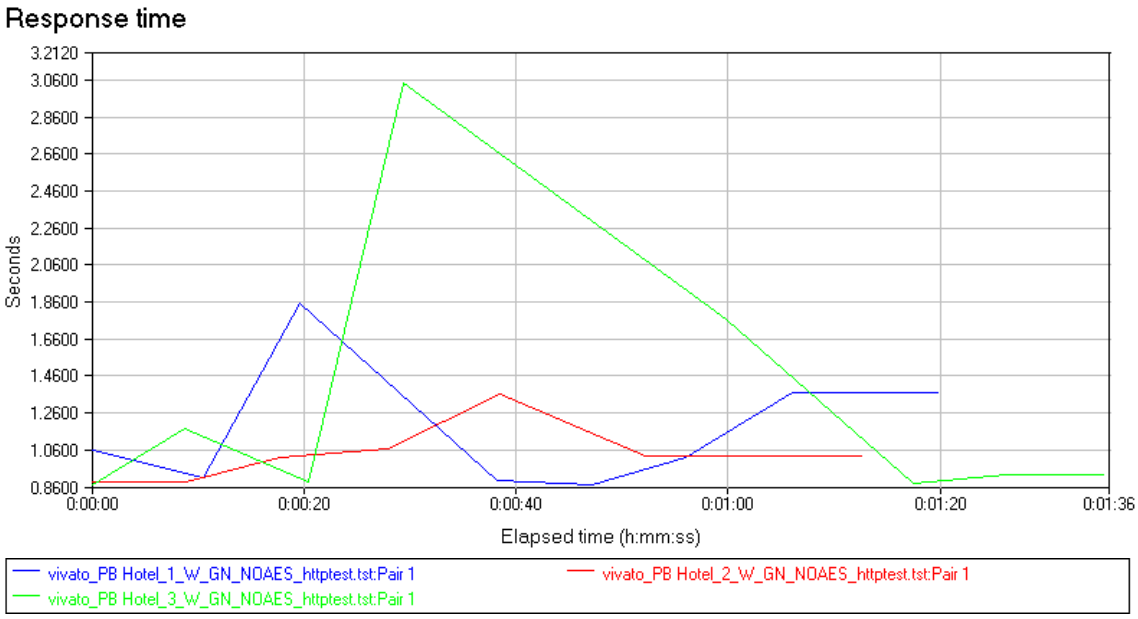

Figure 19. Prachuap Beach Hotel Response Time with GHOSTNet Enabled

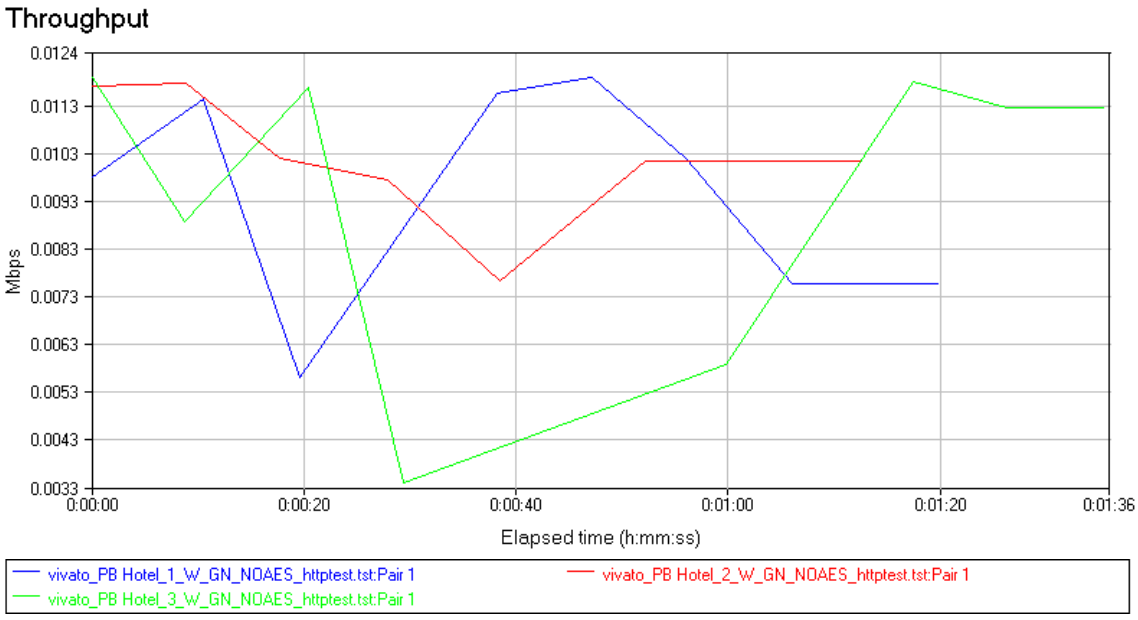

Figure 20. Prachuap Beach Hotel Throughput with GHOSTNet Enabled

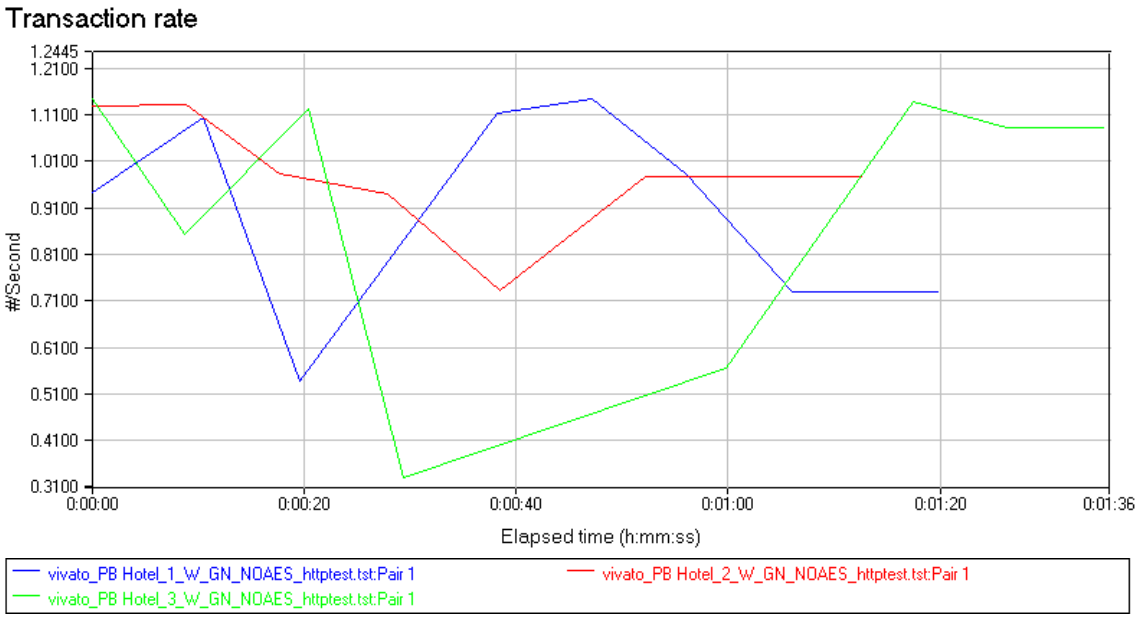

Figure 21. Prachuap Beach Hotel Transaction Rate with GHOSTNet Enabled

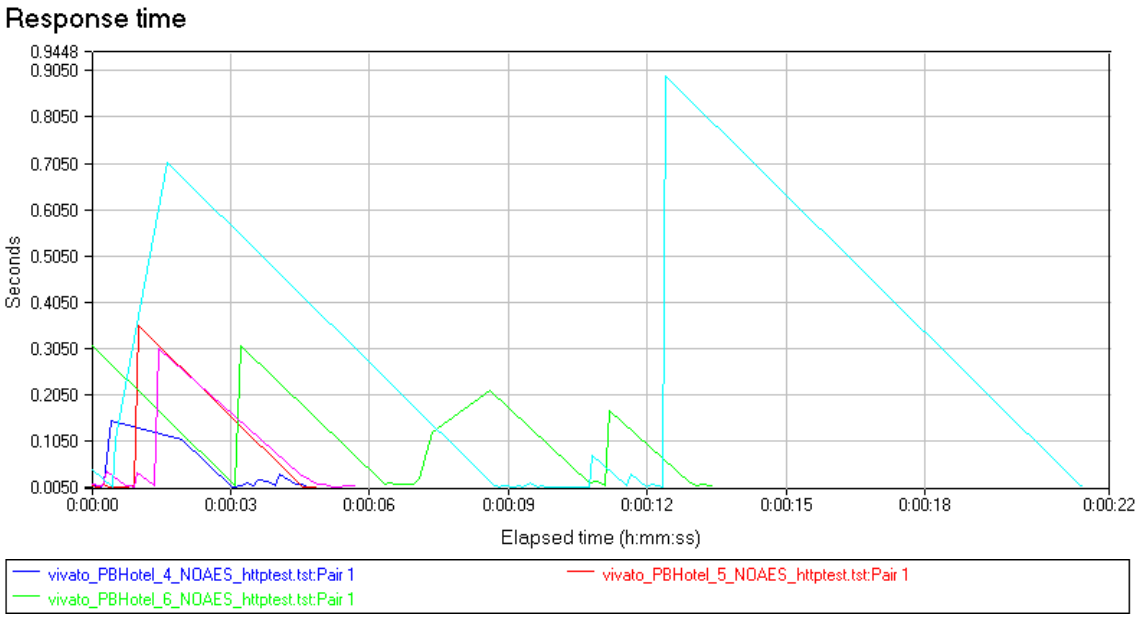

Figure 22. Prachuap Beach Hotel Response Time without GHOSTNet Enabled

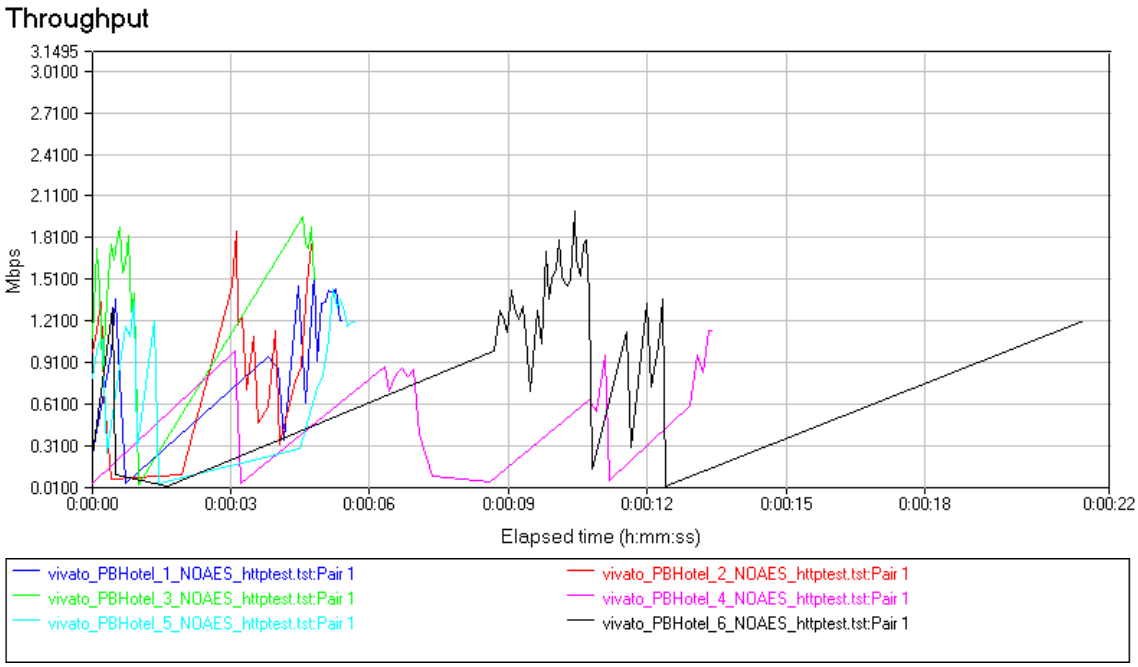

Figure 23. Prachuap Beach Hotel Throughput without GHOSTNet Enabled

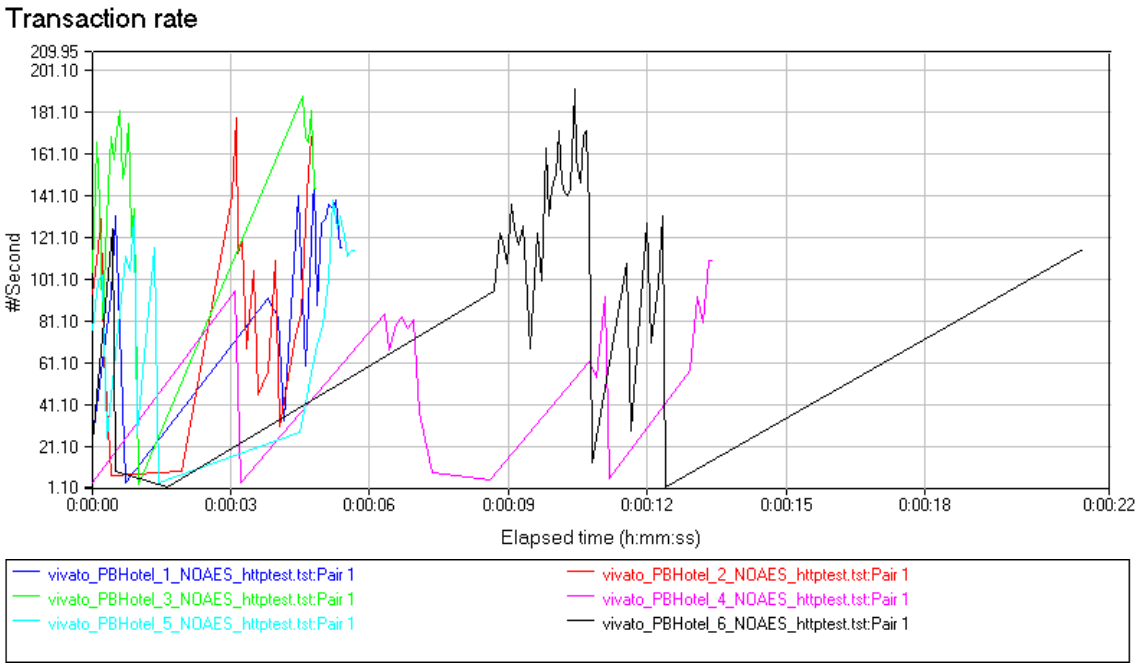

Figure 24. Prachuap Beach Hotel Transaction Rate without GHOSTNet Enabled

c. Test Three: Communications Tower to PCF Underway at 2NM

The third test was conducted between endpoint 1 located at the communications tower (11˚ 47′ 10″ N/099˚ 48′ 34″ E) and endpoint 2 located at the onboard the PCF while underway at 2NM (11˚ 46′ 32″ N /099˚ 48′ 43″ E). Due to time restrictions on the PCFs underway, time testing was conducted on the system without GHOSTNet enabled. Further testing was conducted with GHOSTNet enabled while the PCF was pier side (see test 5).

The average response time without GHOSTNet enabled measured at 2.5207 seconds with a minimum time of .010 seconds, a maximum time of 19.537 seconds and a 95% CI of 3.8978. The average throughput measured at .2363 Mbps with a minimum throughput of .001 Mbps, a maximum throughput of 1.02 Mbps, and a 95% CI of .1952. The average transaction rate measured at 11.9967 seconds with a minimum rate of .051 seconds, a maximum rate of 96.154 seconds, and a 95% CI of 18.7745.

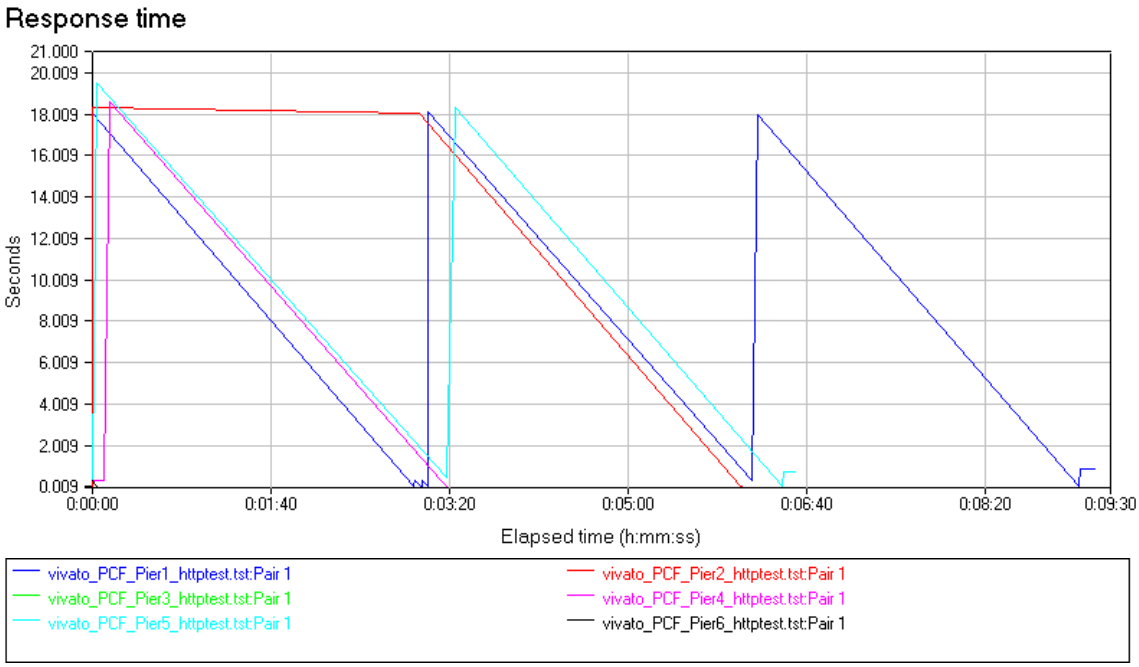

Figure 25. Vivato to PCF Response Time without GHOSTNet Enabled

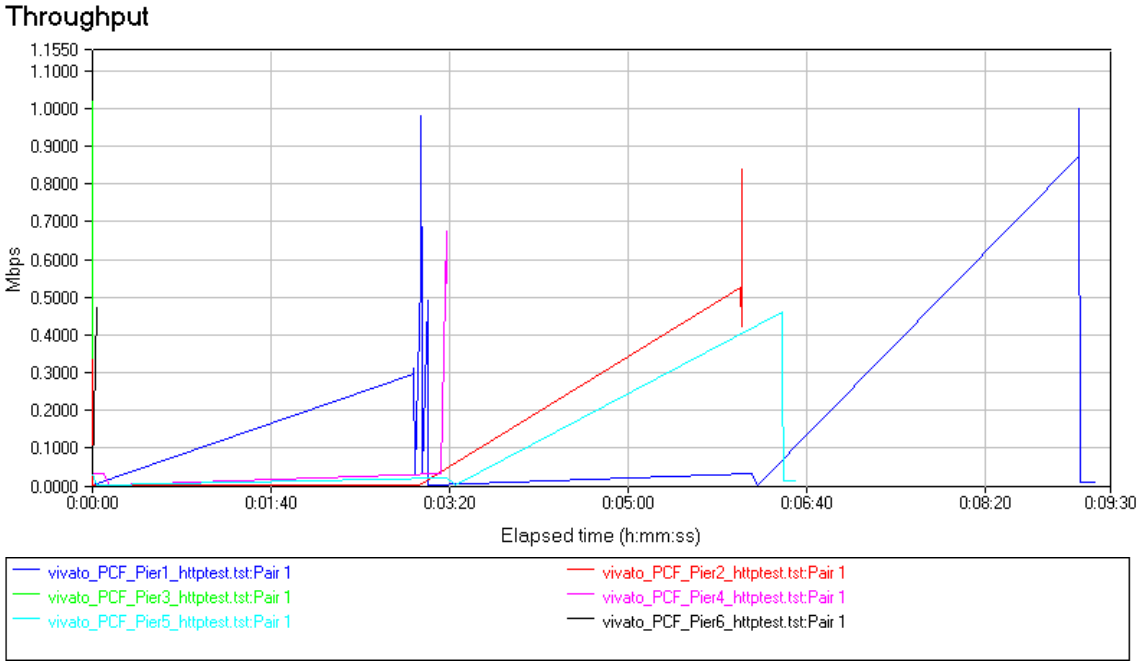

Figure 26. Vivato to PCF Throughput without GHOSTNet Enabled

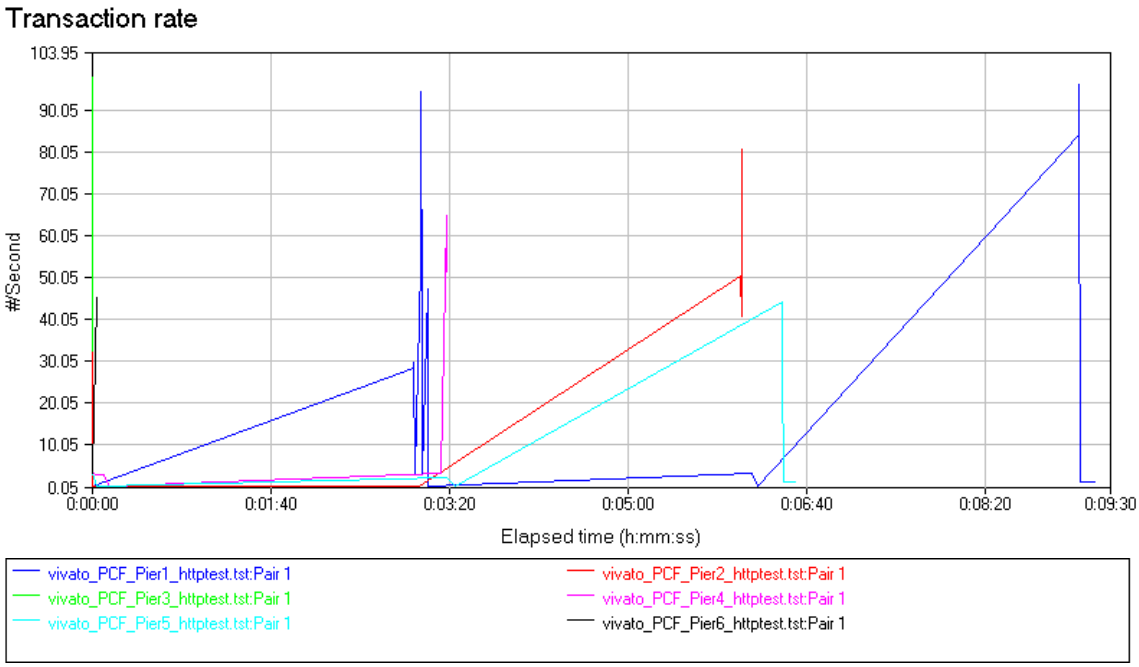

Figure 27. Vivato to PCF Transaction Rate without GHOSTNet Enabled

d. Test Four: Communications Tower to Ao Manao BOQ

The fourth test was conducted between endpoint 1 located at the communications tower (11˚ 47′ 10″ N/099˚ 48′ 34 E) and endpoint 2 located at the Ao Manao BOQ (11˚ 46′ 35″ N/099˚ 47′ 50″ E).

The response time with GHOSTNET enabled measured at 1.0327 seconds with a minimum time of .871, a maximum time of 1.653 and a 95% confidence interval (CI) of .1017. The average throughput measured at .01 Mbps with a minimum of throughput of .006, a maximum throughput of .012, and a 95% CI of .0017. The average transaction rate measured at .9727 seconds with a minimum rate of .605 seconds, a maximum rate of 1.148, and a 95% CI of .1697.

The average response time without GHOSTNet disabled measured at .6698, with a minimum time of .055 seconds, a maximum time of 18.904 seconds and a 95% CI of 1.0118. The average throughput measured at .1377 Mbps, with a minimum throughput of .001 Mbps, a maximum throughput of 2.6 Mbps, and a 95% CI of .2303. The average transaction rate measured at 13.2558 seconds with a minimum rate of .053 seconds, a maximum rate of 243.902 seconds, and a 95% CI of 17.357.

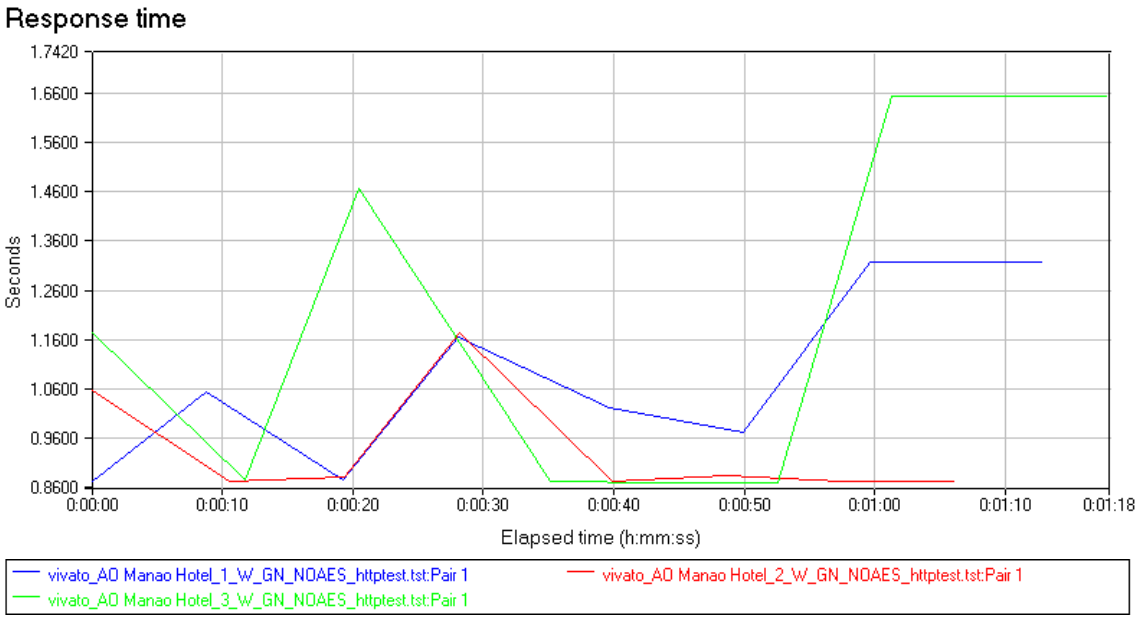

Figure 28. Ao Mano Response with GHOSTNet

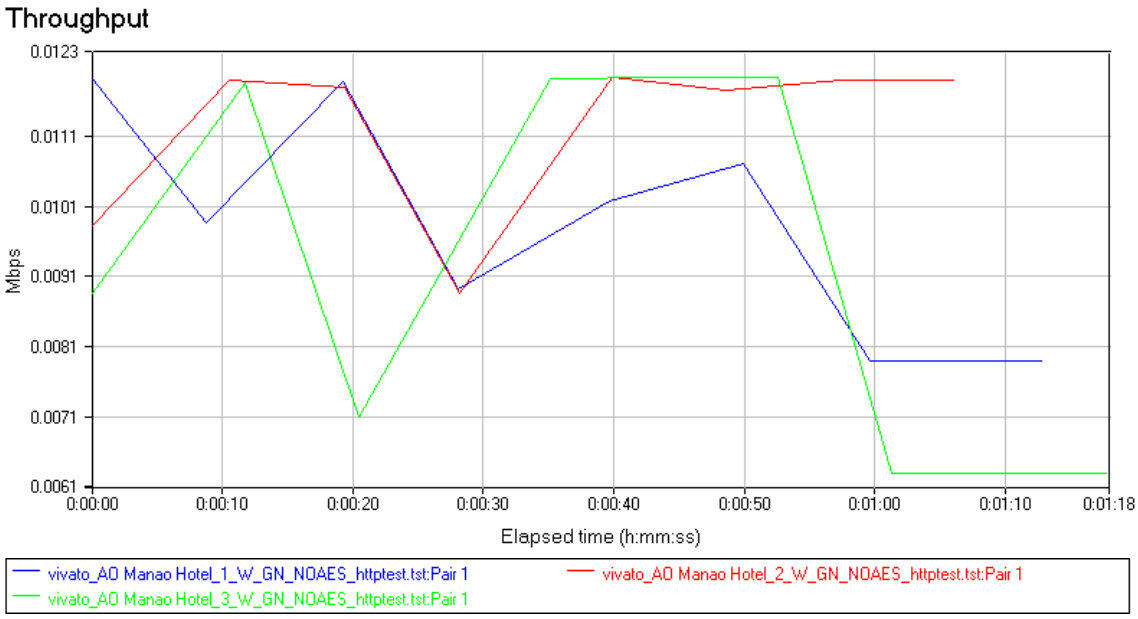

Figure 29. Ao Manao Throughput with GHOSTNET

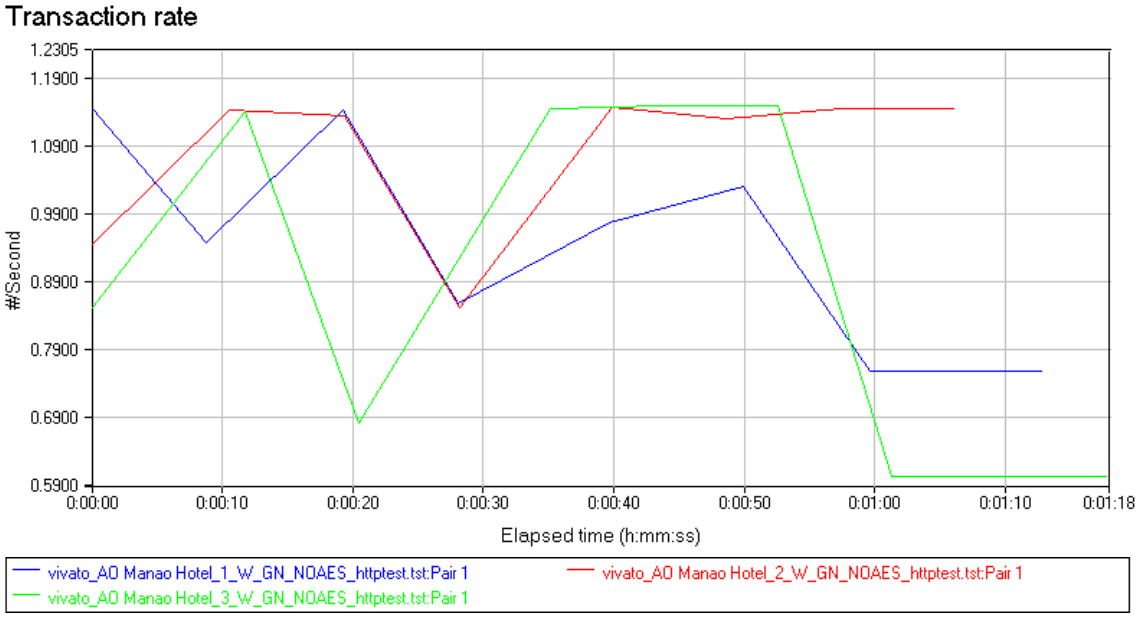

Figure 30. Ao Manao Transaction Rate with GHOSTNet

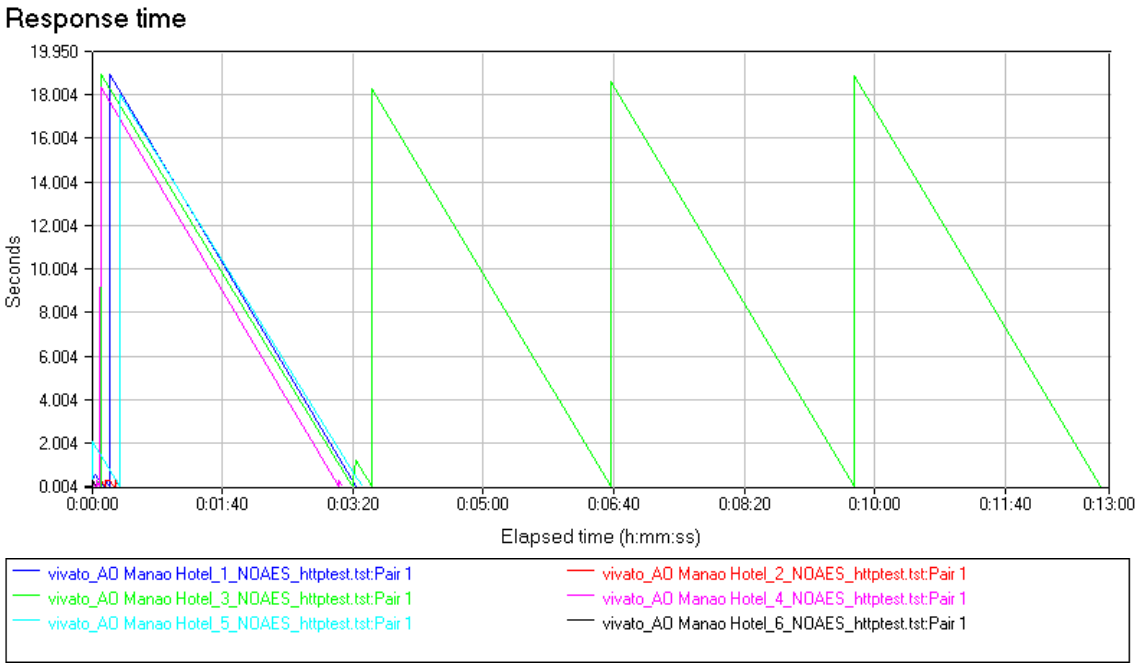

Figure 31. Ao Manao Response Time without GHOSTNet

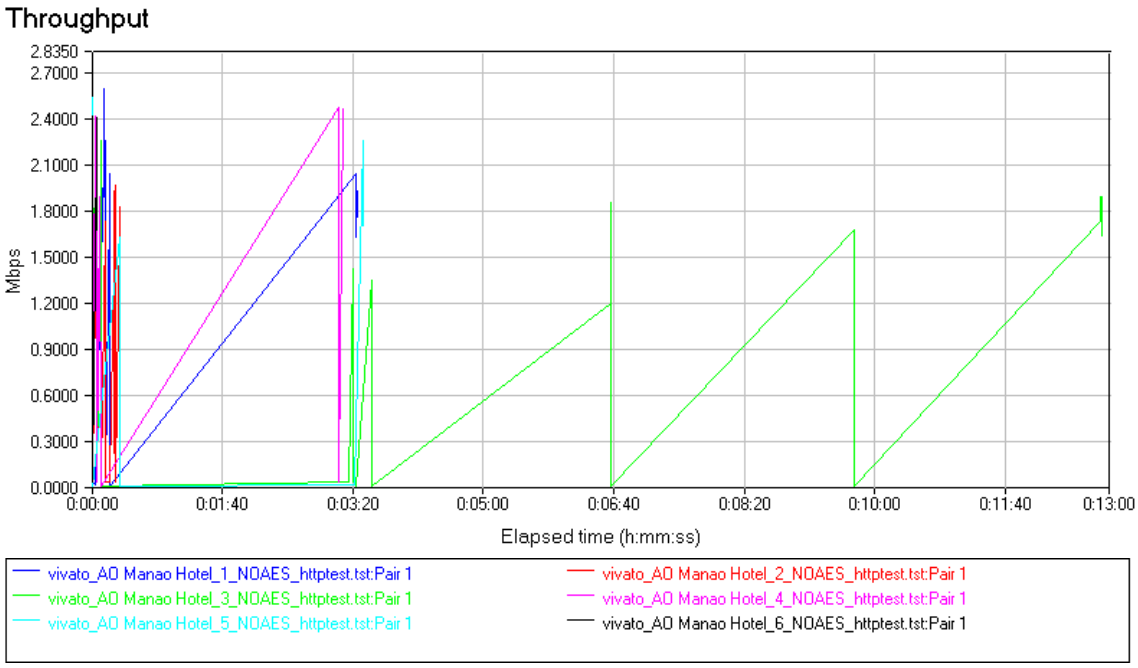

Figure 32. Ao Manao Throughput without GHOSTNet

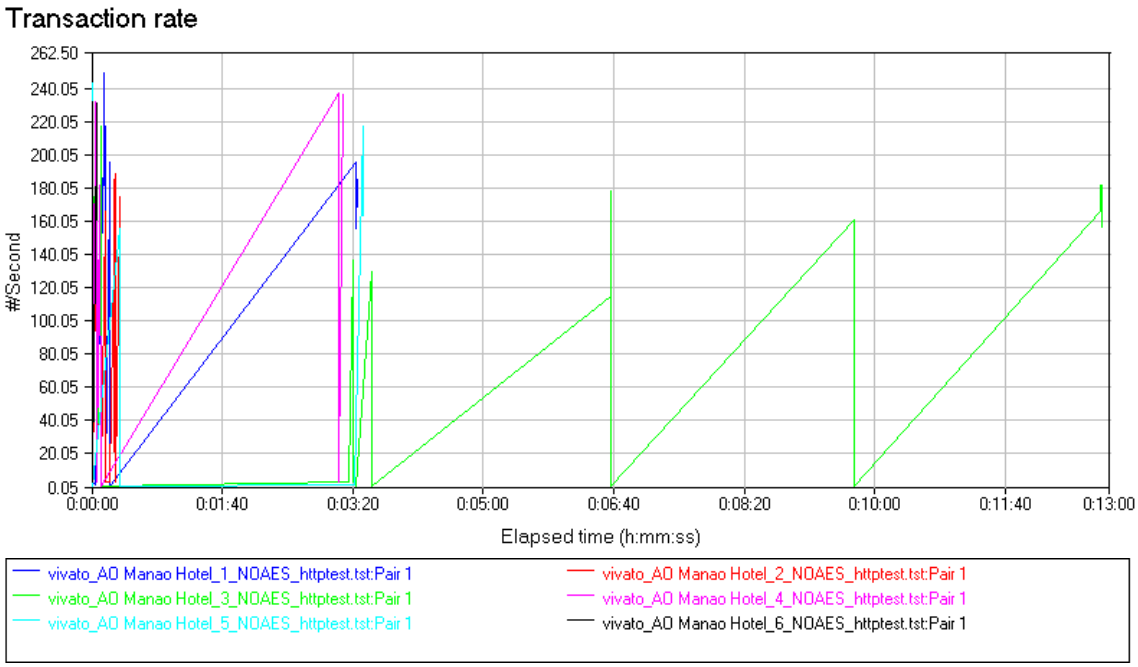

Figure 33. Ao Manao Transaction Rate without GHOSTNet

e. Test Five: Communications Tower to PCF Pier

The fifth test was conducted between endpoint 1 located at the communications tower (11˚ 47′ 10″ N/099˚ 48′ 34″ E) and endpoint 2 located at the onboard the PCF pier side (11˚ 48′ 29″ N/099˚ 48′ 08″ E). The original intent of this testing was to conclude the underway testing with GHOSTNet enabled, but unfortunately due to weather restrictions, the PCF was unable to get underway. It must be noted that this testing was difficult to complete while pier side due to the numerous other vessels that were docked and the interference imposed by their masts.

The average response time with GHOSTNET enabled measured at 11.4763 seconds with a minimum time of .914 seconds, a maximum time of 38.854 seconds and a 95% CI of 32.4613. The average throughput measured at .001 Mbps with a

minimum of throughput of .000 Mbps, a maximum throughput of .011 Mbps, and a 95% CI of .0037. The average transaction rate measured at .1060 seconds with a minimum rate of .027 seconds, a maximum rate of 1.094, and a 95% CI of .3590.

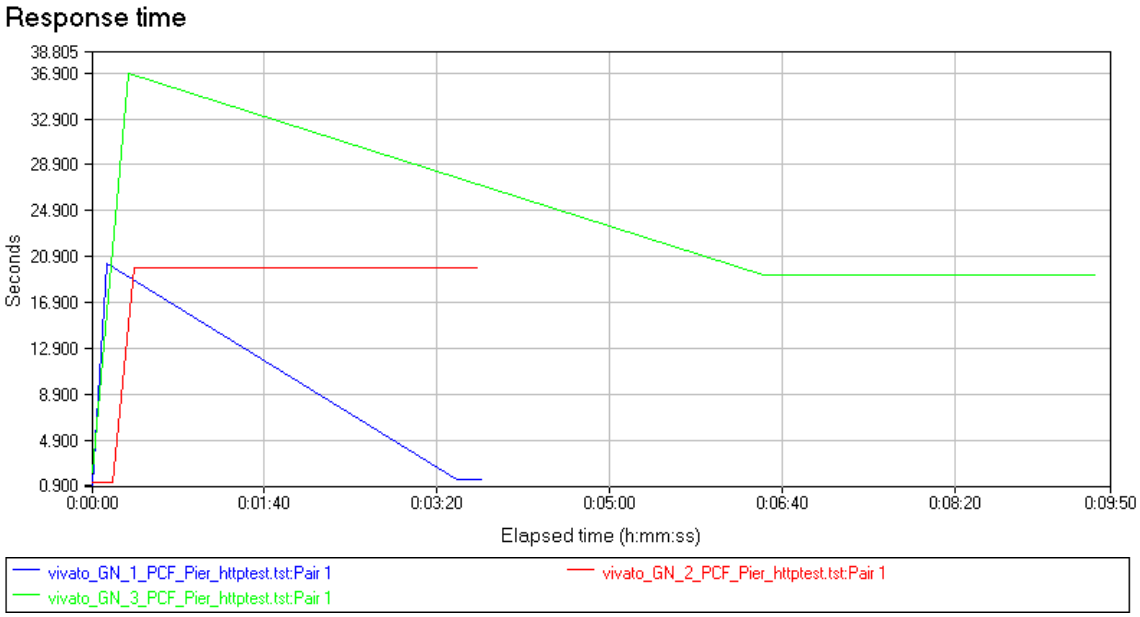

Figure 34. Vivato to Pier Response Time with GHOSTNet enabled

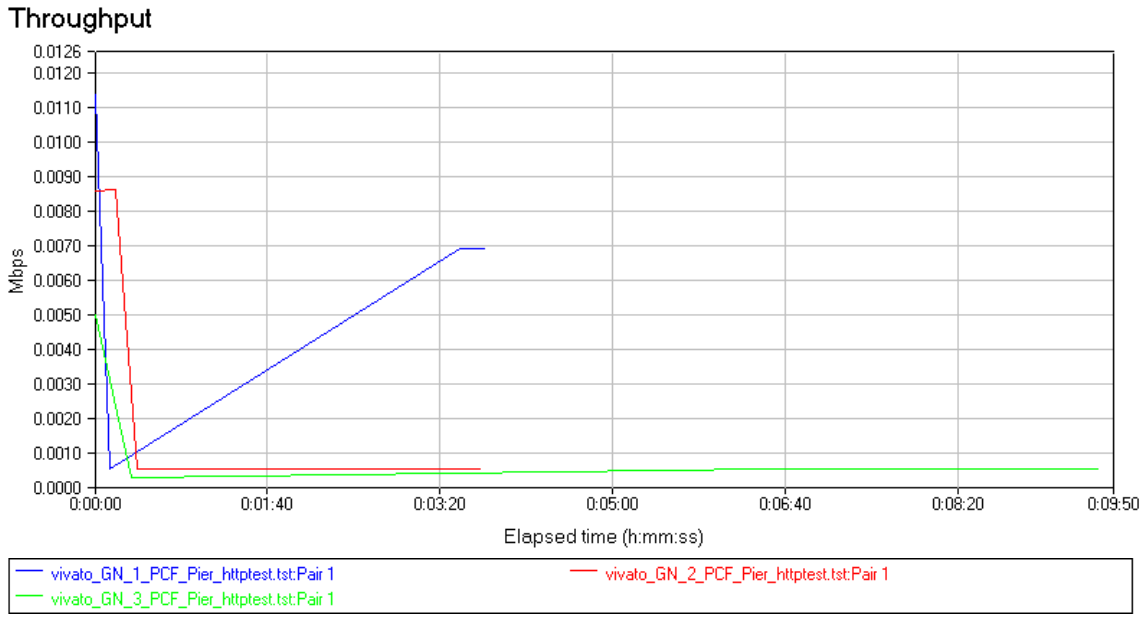

Figure 35. Vivato to Pier Throughput with GHOSTNet enabled

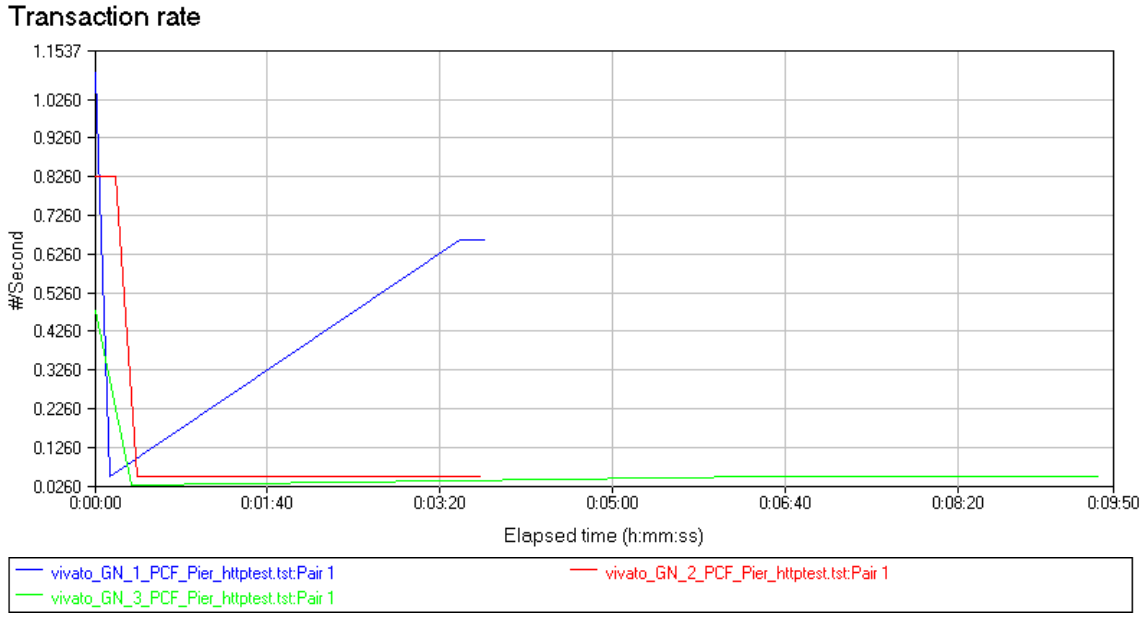

Figure 36. Vivato to Pier Transaction Rate with GHOSTNet enabled

4. Conclusions from Ao-Manao Thailand

As the data shows, the network experienced slow response times, low throughput rates, slow transaction times, and excessive latency when GHOSTNet was enabled. Due to the slowness, it was determined that it may be coming from GHOSTNet connecting to the servers located in New Haven, Connecticut, or Greensboro, North Carolina. To test this theory a GHOSTNet server was established in Monterey, California. Below are the results of the testing conducted in Monterey.

Server	Latency (ms)	Upload (kb/s)	Download (kb/s)
San Francisco	16	335	2364
Chicago	60	347	1206
Parsippany	55	306	1818
Toronto	94	348	1948

Table 3. Speed and Latency Test with Local GHOSTNet Server Connected

Server	Latency (ms)	Upload (kb/s)	Download (kb/s)
San Francisco	8	333	1589
Chicago	47	334	1155
Parsippany	71	358	1890
Toronto	94	335	1863

Table 4. Speed and Latency Test with No GHOSTNet Server Connected

Server	Latency (ms)	Upload (kb/s)	Download (kb/s)
San Francisco	126	918	178
Chicago	119	917	220
Parsippany	279	625	160
Fort Worth	120	765	184

Table 5. Speed and Latency Test with GHOSTNet connected to New Haven, CT Server

a. Test One: Baseline of System in Monterey, CA

The first test was conducted between endpoint 1 located and endpoint 2 to baseline the system. Both endpoints were connected wirelessly to the same network (192.168.1.1), and the test was conducted with GHOSTNet disabled and enabled.

The average response time with GHOSTNET enabled measured at .0310 seconds with a minimum time of .023 seconds, a maximum time of .159 seconds and a 95% CI of 0025. The average throughput measured at 2.4042 Mbps with a minimum throughput of .476 Mbps, a maximum throughput of 3.025 Mbps, and a 95% CI of .2047. The average transaction rate measured at 32.3963 seconds with a minimum rate of 6.289 seconds, a maximum rate of 40.00, and a 95% CI of 2.4867.

The average response time with GHOSTNet disabled measured at .0225 with a minimum time of .011 seconds, a maximum time of .055 seconds and a 95% CI of .0025. The average throughput measured at 4.4682 Mbps with a minimum throughput of .0635 Mbps, a maximum throughput of 6.874 Mbps, and a 95% CI of .3435. The average transaction rate measured at 49.5712 seconds with a minimum rate of 8.403 seconds, a maximum rate of 90.909 seconds, and a 95% CI of 4.5412.

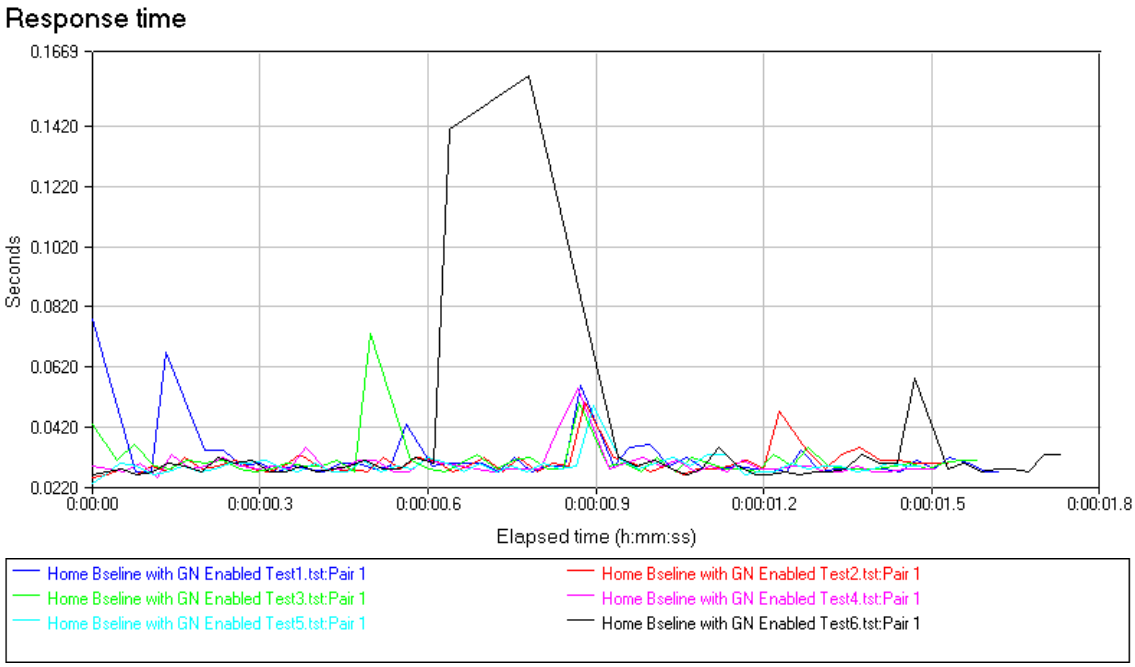

Figure 37. Home Baseline Response Time with GHOSTNet Enabled

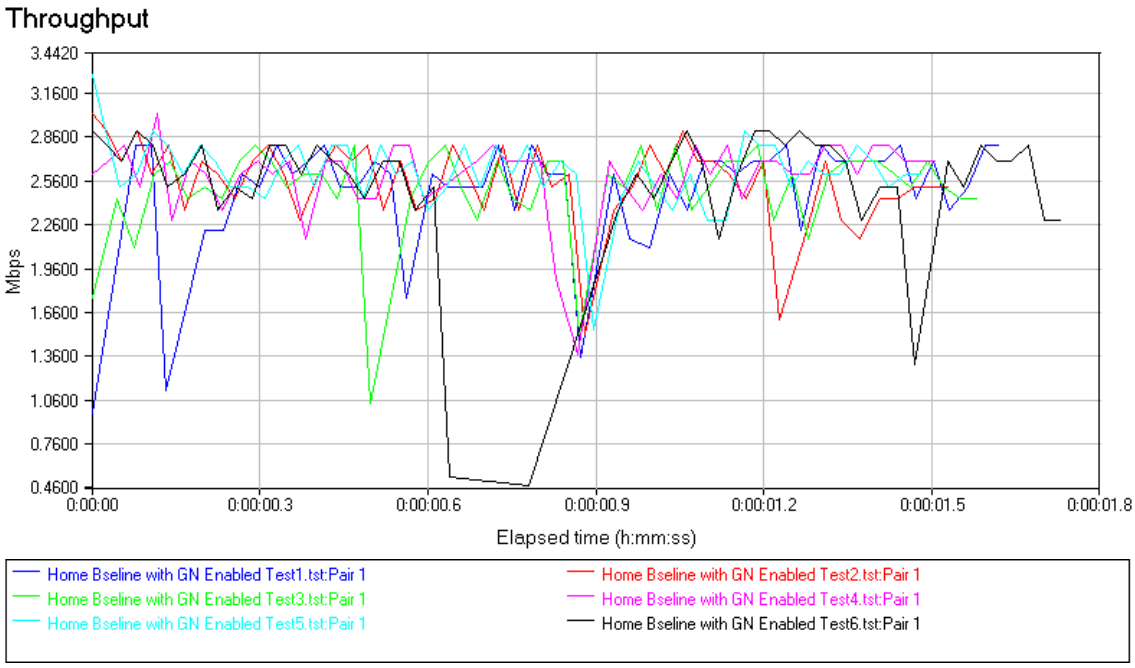

Figure 38. Home Baseline Throughput with GHOSTNet Enabled

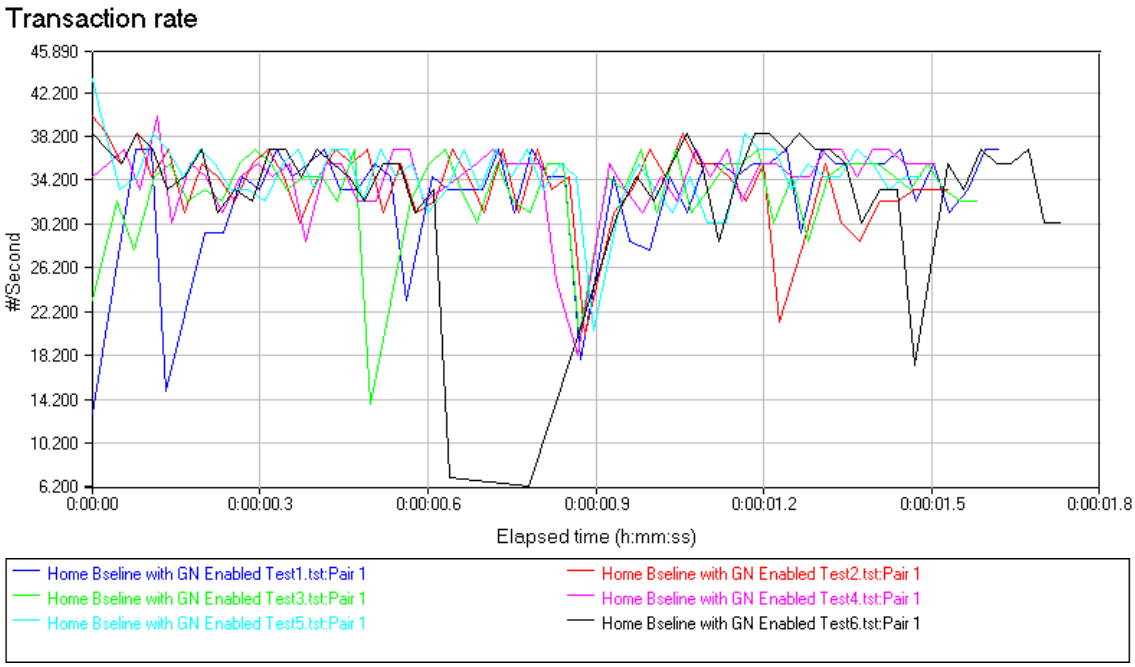

Figure 39. Home Baseline Transaction Rate with GHOSTNet Enabled

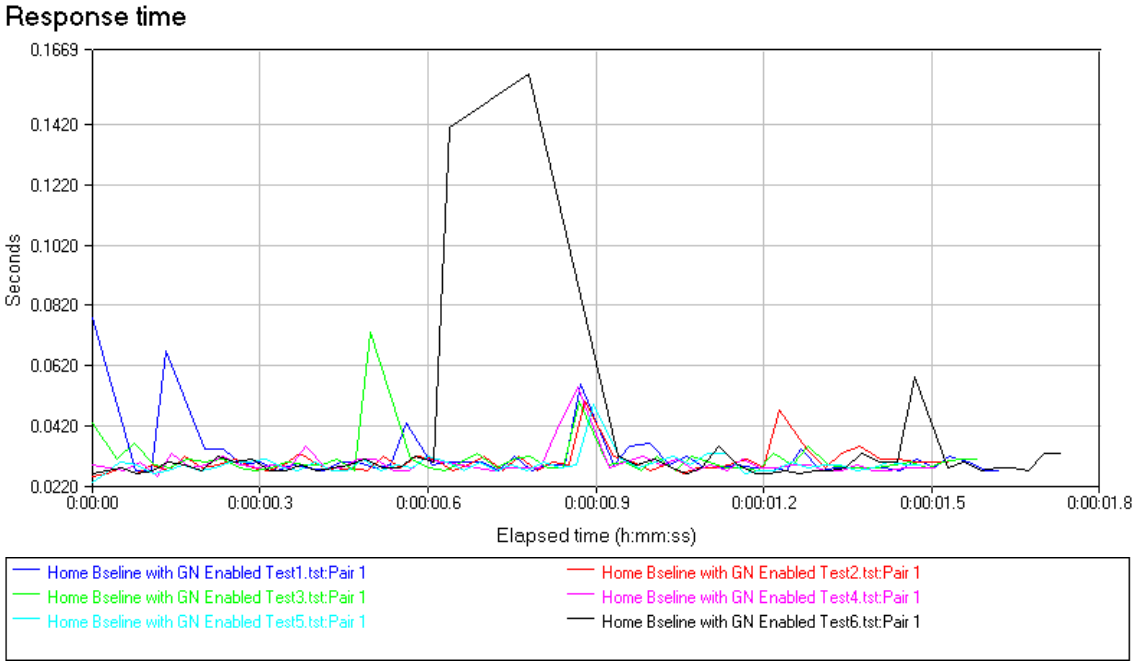

Figure 40. Home Baseline Response Rate without GHOSTNet Enabled

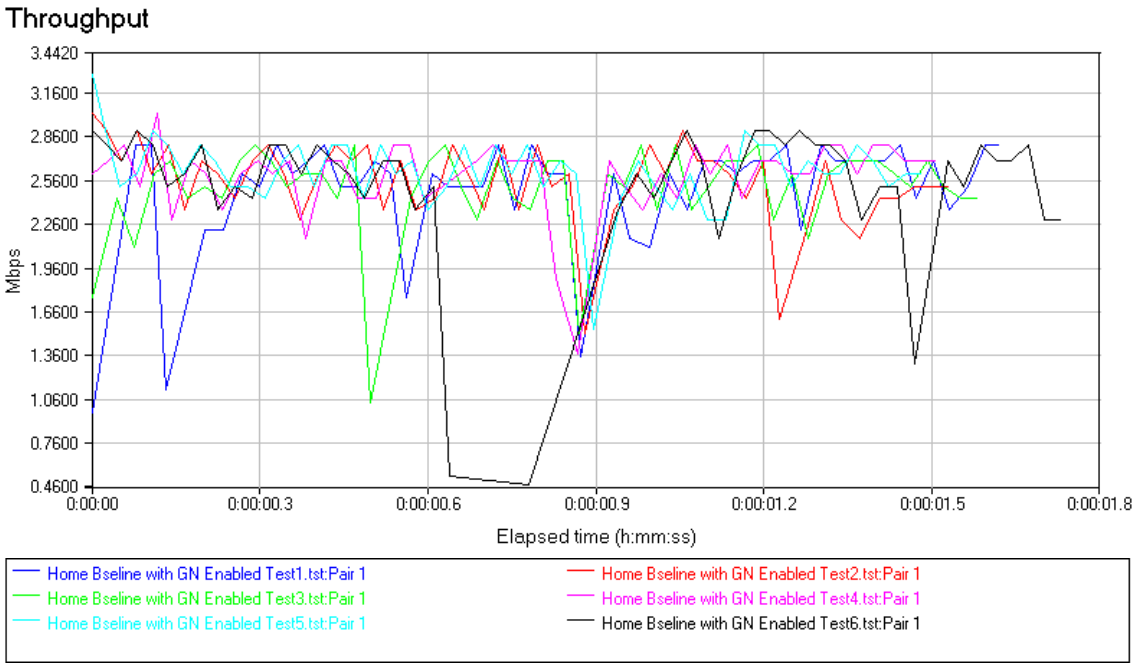

Figure 41. Home Baseline Throughput without GHOSTNet Enabled

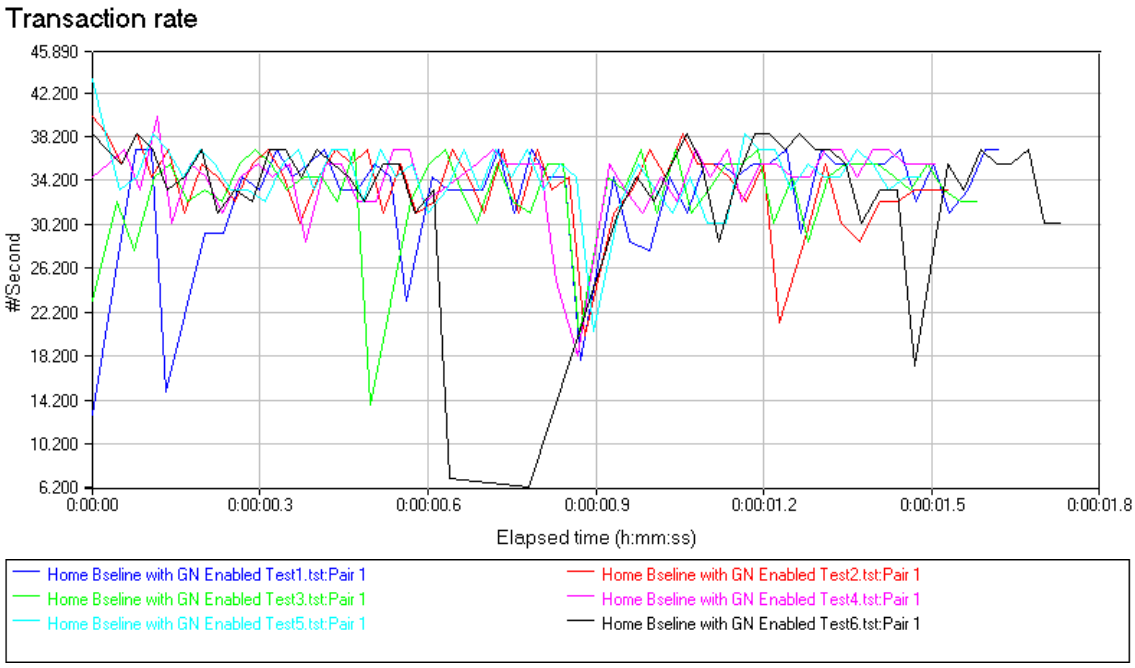

Figure 42. Home Baseline Transaction Rate without GHOSTNet Enabled

b. Test Two: Testing Between USCG Station Monterey Bay and Local GHOSTNet Server

The second test was conducted between endpoint 1 located at the Coast Guard Station (36°36′33″N 121°53″49″W and endpoint 2 colocated with the local GHOSTNet Server (36°37′54″N 121°48′24″W). Endpoint 1 was wired via Ethernet to the Coast Guard Stations Network, endpoint 2 was associated wirelessly to the network colocated with the local GHOSTNet server, and the test was conducted with GHOSTNet enabled.

The average response time measured at .923 seconds with a minimum time of .0237 seconds, a maximum time of 84.589 seconds and a 95% CI of .69. The average throughput measured at .252 Mbps with a minimum throughput of .017 Mbps, a maximum throughput of .319 Mbps, and a 95% CI of .029. The average transaction rate measured at 1.632 seconds with a minimum rate of .012 seconds, a maximum rate of 4.129 seconds, and a 95% CI of .386.

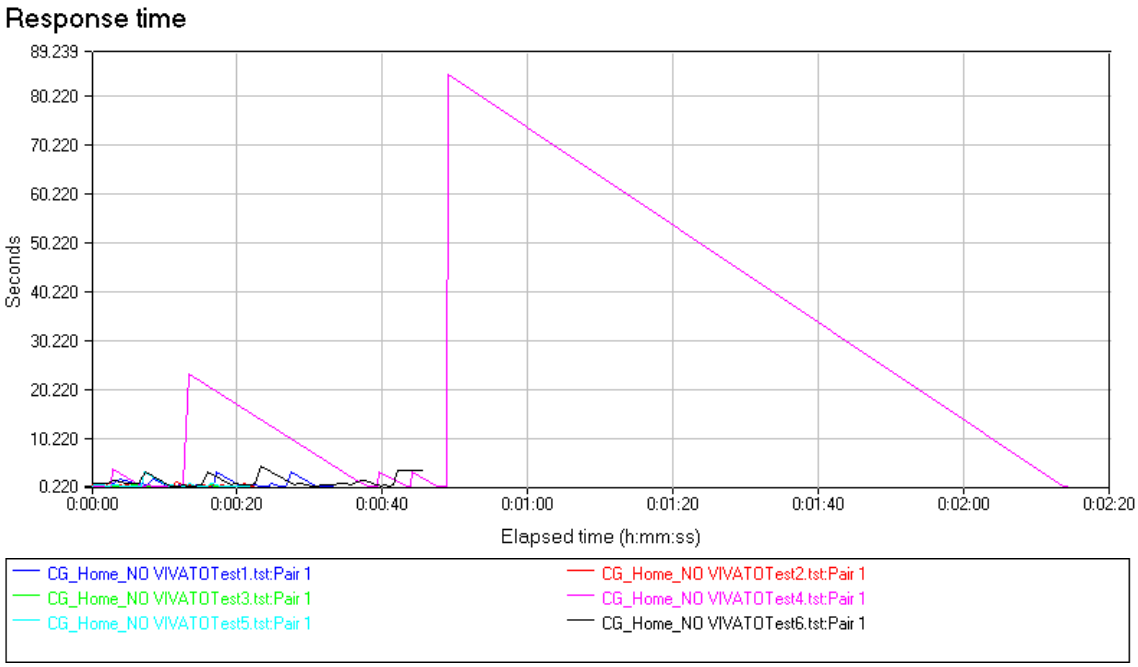

Figure 43. Coast Guard Station to Local GHOSTNet Server Response Time

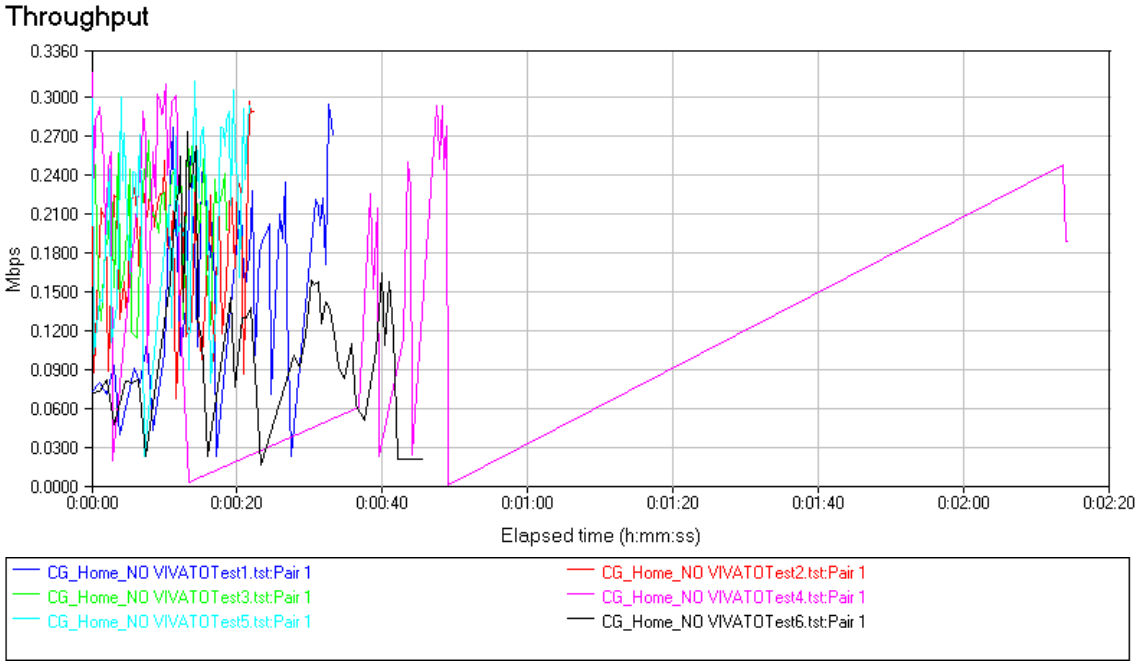

Figure 44. Coast Guard Station to Local GHOSTNet Server Throughput

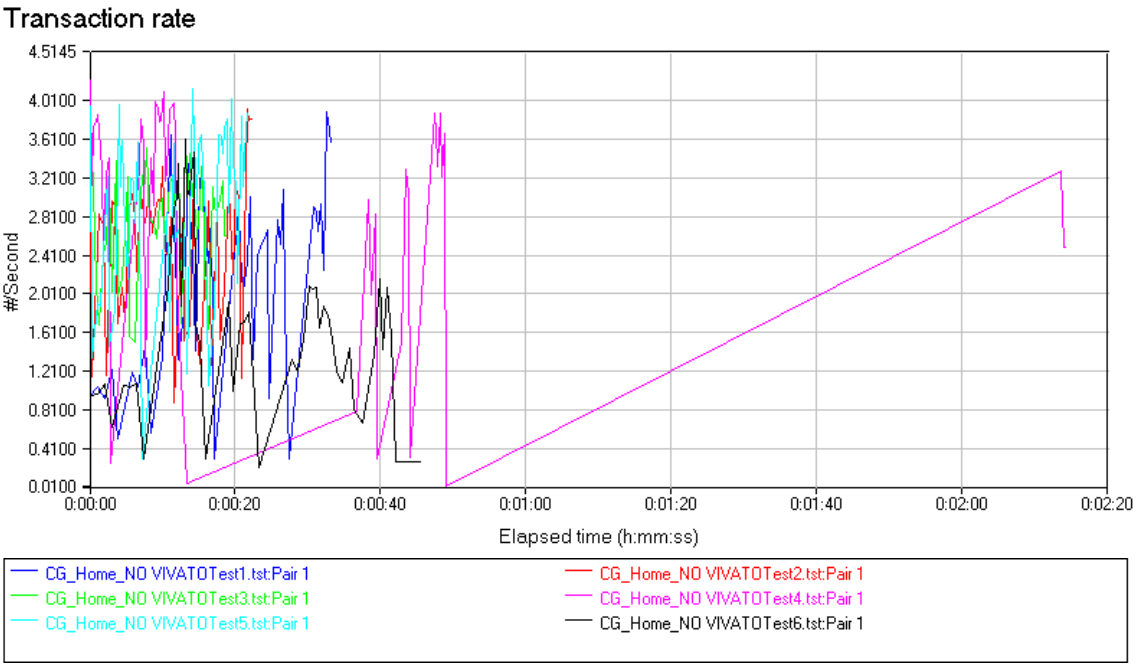

Figure 45. Coast Guard Station to Local GHOSTNet Server Transaction Rate

c. Test Three: Voice Test

The third test was conducted between endpoint 1 located at the Coast Guard Station (36°36’33”N 121°53”49”W and endpoint 2 colocated with the local GHOSTNet Server (36°37’54”N 121°48’24”W). Endpoint 1 was wired via Ethernet to the Coast Guard Stations Network, endpoint 2 was associated wirelessly to the network colocated with the local GHOSTNet server, and the test was conducted with GHOSTNet enabled. The purpose of this test was to measure the quality of voice data being sent across the network.

The average response time measured at .24 seconds with a minimum time of .203 seconds, a maximum time of .361 seconds and a 95% CI of .025. The average throughput measured at .015 Mbps with a minimum throughput of .012

Mbps, a maximum throughput of .019 Mbps, and a 95% CI of .0017. The average transaction rate measured at 4.27 seconds with a minimum rate of 2.767 seconds, a maximum rate of 4.934 seconds, and a 95% CI of .447.

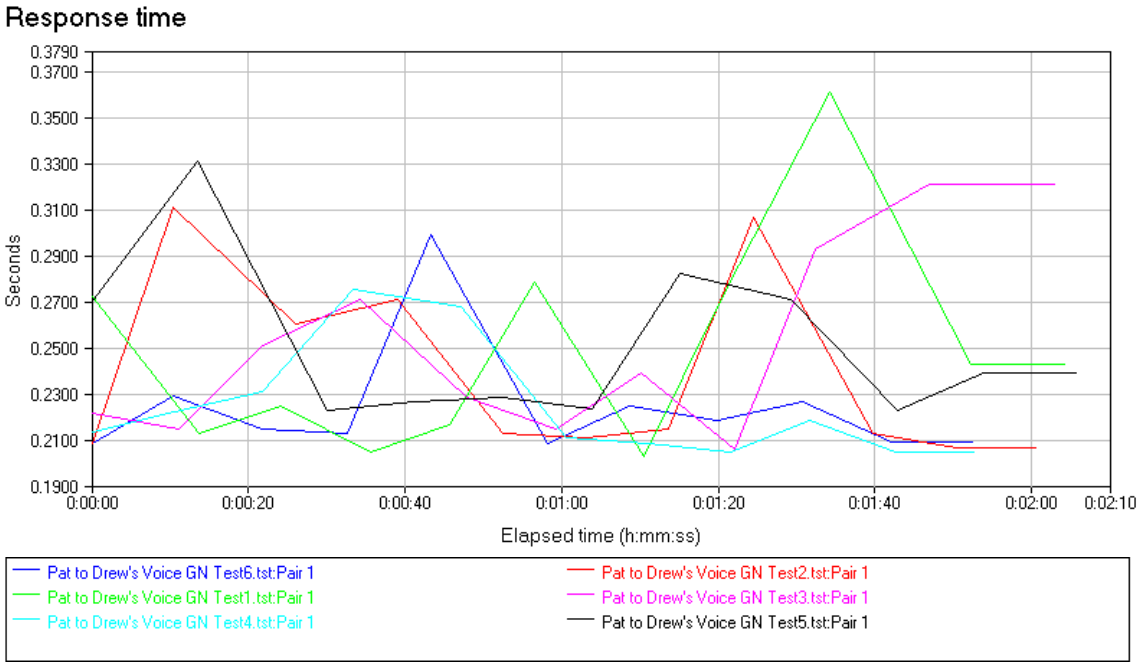

Figure 46. Voice Test Response Time with GHOSTNet Connected Through the Local Server

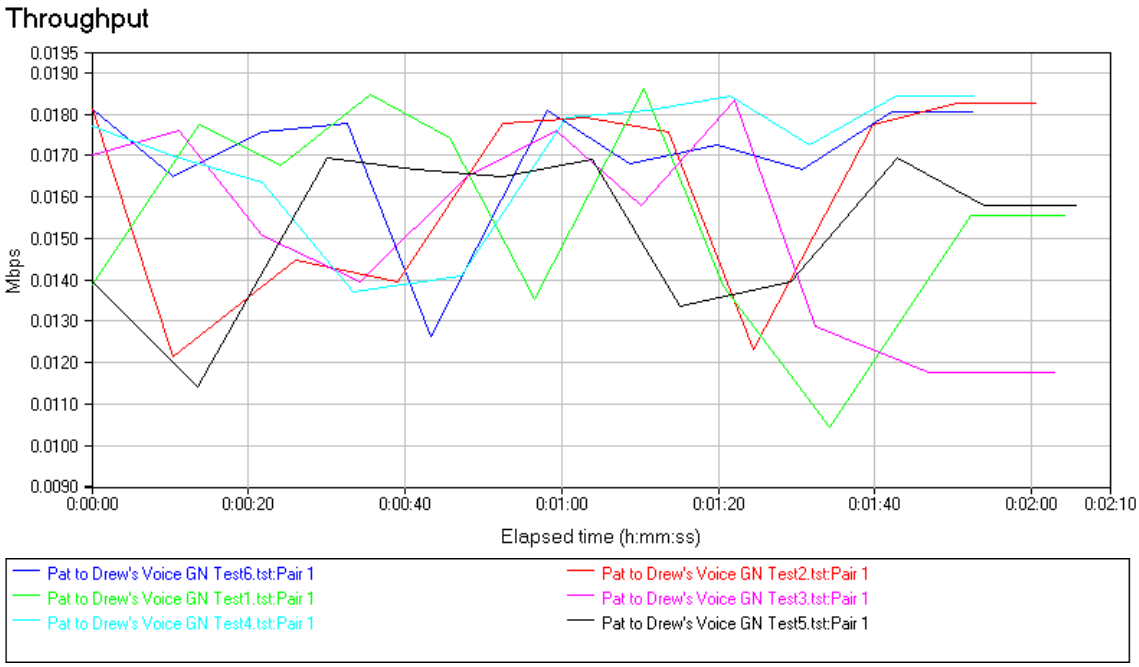

Figure 47. Voice Test Throughput with GHOSTNet Connected Through the Local Server

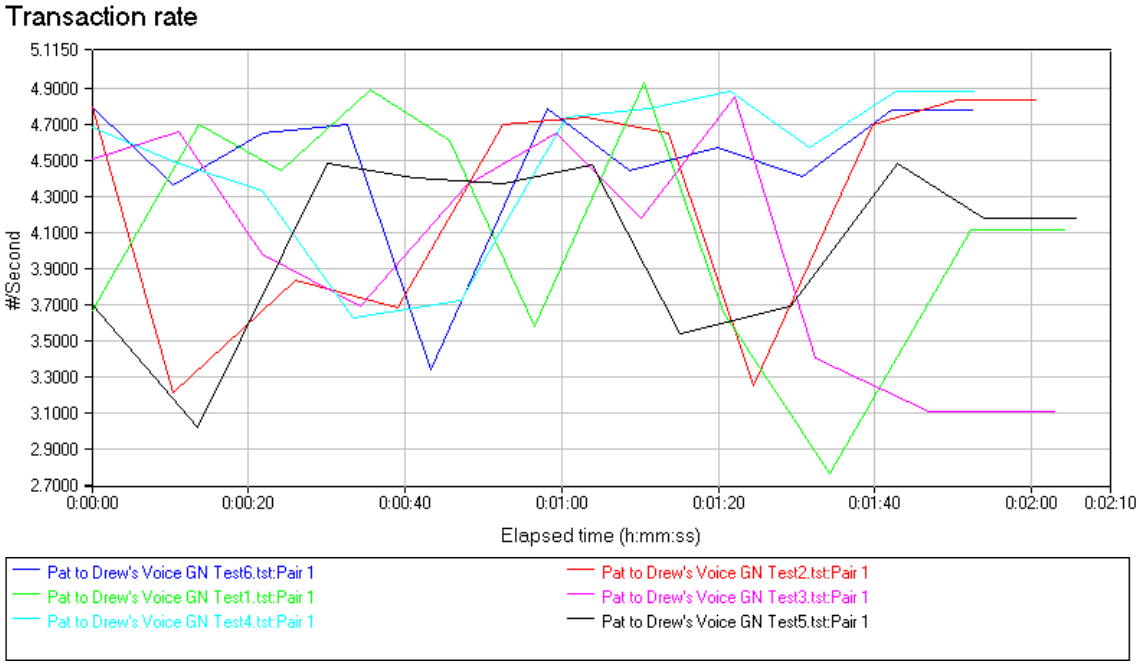

Figure 48. Voice Test Transaction Rate with GHOSTNet Connected Through the Local Server

d. Test Four: Video Test

The fourth test was conducted between endpoint 1 located at the Coast Guard Station (36°36’33”N 121°53”49”W and endpoint 2 colocated with the local GHOSTNet Server (36°37’54”N 121°48’24”W). Endpoint 1 was wired via Ethernet to the Coast Guard Stations Network, endpoint 2 was associated wirelessly to the network colocated with the local GHOSTNet server, and the test was conducted with GHOSTNet enabled. The purpose of this test was to measure the quality of video being sent across the network. Using IXChariot a 7.96 mb file was sent between the two endpoints.

The average throughput measured at .064 Mbps with a minimum throughput of .049 Mbps, a maximum throughput of .062 Mbps. The average percent of loss data was .2.

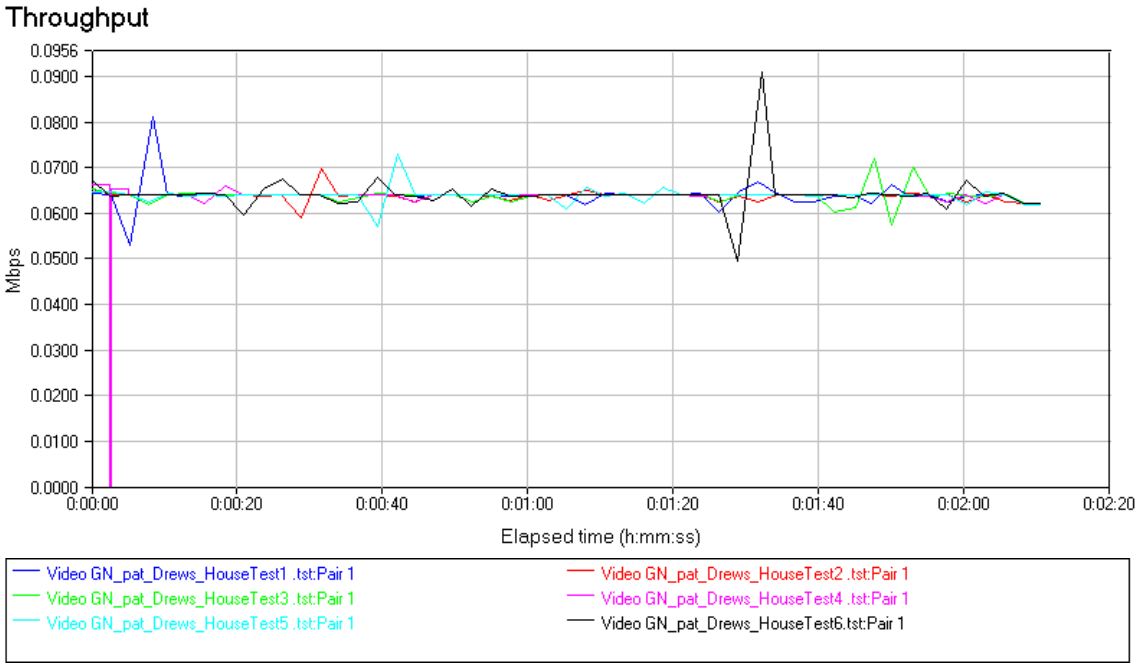

Figure 49. Video Test Throughput with GHOSTNet Connected Through the Local Server

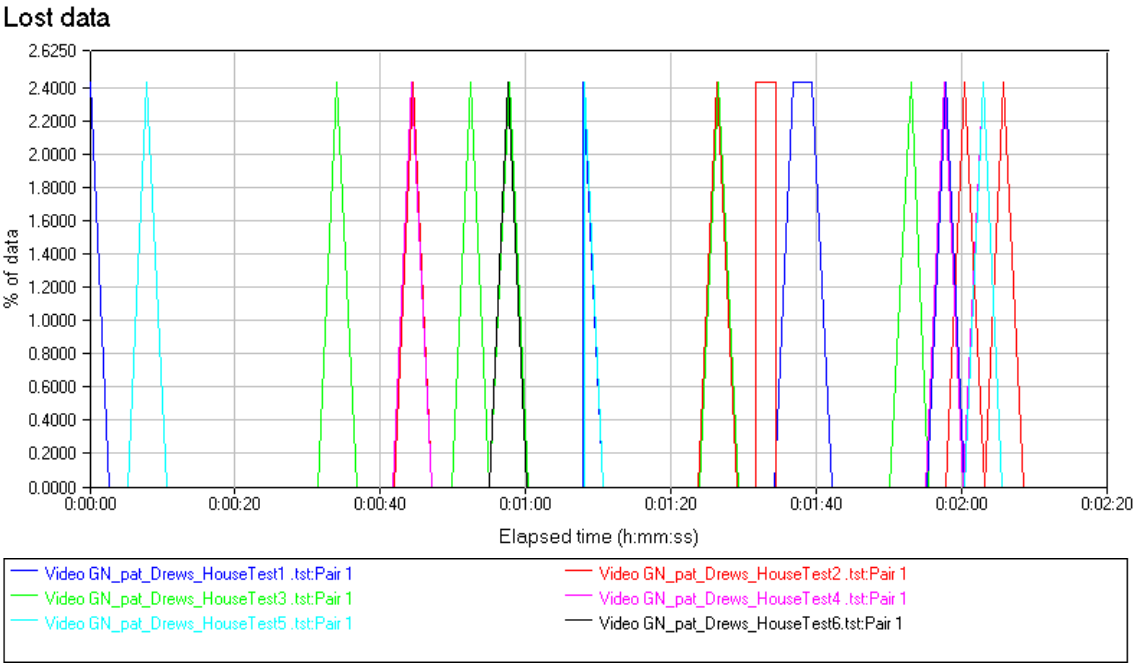

Figure 50. Video Test Lost Data with GHOSTNet Connected Through the Local Server

V. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

In the National Security Strategy of the United States of America, former President Bush outlines priorities to enhance the transformation of key institutions, to include improving the capacity of agencies to plan, prepare, coordinate, integrate, and execute responses covering the full range of crisis contingencies and long-term challenges.[14] Additionally, President Bush points out that in order for domestic agencies to assist in keeping the United States secure, we must get better at interagency integration at home and abroad.[15] GHOSTNet will enable DoD, Homeland Security, and local Law enforcement personnel to form networks to share information securely. The ideal placement GHOSTNet servers would be in a Maritime Headquarters with maritime Operations Center (MHQ with MOC). This would allow ships to dynamically connect to the command center and display, route, print, or brief any mission within the AOR prior to execution. MHQ staff personnel would be full apprised of any situation and could have the capability to watch any mission from Headquarters. Additionally, GHOSTNet could become a driving factor to assist the United States military transitions to Network-Centric Warfare (NCW). NCW is the Tactic, Techniques,

14 George W. Bush, "The National Security Strategy of the United States of America," March 2006, 45, available from http://www.globalsecurity.org/military/library/policy/national/nss-060316.htm (accessed 30 April 2009).

15 Ibid., 46.

Procedures (TTPs) that a networked force could employ to create an influential advantage to the deployed force.

1. Key Findings

Throughout the experiments conducted in support of this thesis, it was noted the response rate, throughput, and transaction rate of the network when GHOSTNet was enabled were significantly reduced, but not to the capacity to render the network useless. Of note was the latency in the data when GHOSTNet was connected. Users experienced an average latency of 161 ms, and are more likely to disconnect the secure connection and pass traffic in the open at the risk of data being compromised as was seen during FEX V in Thailand. To reduce the latency experienced, a local GHOSTNet server was established. During initial testing latency was reduced from an average of 161 ms to an average 56.25 ms. This significant decrease is comparable to the average latency, 55 ms, a user might experience through a non-secure internet connection.

B. CONCLUDING REMARKS

1. Future Research

There are numerous research opportunities that can be conducted between a GHOSTNet enabled network and end user devices that could ultimately be utilized by Command and Control cells to pass data securely and assist in the transformation into the era of Network-Centric Warfare. Following are just two examples.

a. Mobile Communication Devices

Technology in cellular phones has been greatly enhanced, and with the advent of the Blackberry®, iPhone®, and the gPhone®, users are continually networked.

Future ideas include loading and testing the GHOSTNet architecture on mobile communication devices to transmit and send data securely across the network.

b. Local GHOSTNet Server with Anonymization

Although a local GHOSTNet server was established and tested in the Monterey area, further testing needs to be conducted on the latency once a proxy server has been added to the network architecture to provide anonymization for units that need this type of service.

2. Summary

The GHOSTNet application allows additional flexibility in securely connecting diverse units on an as-needed basis. The application also allows the Commanding Officer of units deployed conducting MIO to securely view, pass data, and communicate with their Boarding Officer and upper echelon chain of command via networks. This added capability could prove invaluable in the amount and quality of the intelligence gathered during this type of operations realizing the goals of Network-Centric Warfare. Different network designs that incorporate localized GHOSTNet servers allow for performance measures to nearly match those of traditional Internet based open networks so that a trade-off between performance and security need not be made.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX: GHOSTNET SERVER AND CLIENT CONFIGURATION FILES

A. WINDOWS XP SERVER CONFIGURATION FILE

```
##############################################
# Sample OpenVPN 2.0 config file for         #
# multi-client server.                       #
#                                            #
# This file is for the server side           #
# of a many-clients <-> one-server           #
# OpenVPN configuration.                     #
#                                            #
# OpenVPN also supports                      #
# single-machine <-> single-machine          #
# configurations (See the Examples page      #
# on the web site for more info).            #
#                                            #
# This config should work on Windows         #
# or Linux/BSD systems.  Remember on         #
# Windows to quote pathnames and use         #
# double backslashes, e.g.:                  #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                            #
# Comments are preceded with '#' or ';'      #
```

```
#################################################

# Which local IP address should OpenVPN
# listen on? (optional)
local 192.168.1.109
# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194
mssfix 1400
# TCP or UDP server?
;proto tcp
proto udp
# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
```

```
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
dev tap
#dev tun
#tls-server
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap
# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
```

```
# and each of the client certificates.
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ifconfig 10.8.0.1 255.255.252.0
tls-server
ca Home_GN_Keys/ca.crt
cert Home_GN_Keys/VPN.crt
key Home_GN_Keys/VPN.key  # This file should be kept
secret
# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh1024.pem 1024
# Substitute 2048 for 1024 if you are using
# 2048 bit keys.
dh Home_GN_Keys/dh1024.pem
# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
# the rest will be made available to clients.
# Each client will be able to reach the server
# on 10.8.0.1. Comment this line out if you are
# ethernet bridging. See the man page for more info.
server 10.8.0.0 255.255.252.0
```

```
# Maintain a record of client <-> virtual IP address
# associations in this file.  If OpenVPN goes down or
# is restarted, reconnecting clients can be assigned
# the same virtual IP address from the pool that was
# previously assigned.
ifconfig-pool-persist ipp.txt
# Configure server mode for ethernet bridging.
# You must first use your OS's bridging capability
# to bridge the TAP interface with the ethernet
# NIC interface.  Then you must manually set the
# IP/netmask on the bridge interface, here we
# assume 10.8.0.4/255.255.255.0.  Finally we
# must set aside an IP range in this subnet
# (start=10.8.0.50 end=10.8.0.100) to allocate
# to connecting clients.  Leave this line commented
# out unless you are ethernet bridging.
;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50
10.8.0.100
# Configure server mode for ethernet bridging
# using a DHCP-proxy, where clients talk
# to the OpenVPN server-side DHCP server
# to receive their IP address allocation
# and DNS server addresses.  You must first use
# your OS's bridging capability to bridge the TAP
```

```
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge
# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
;push "route 10.80.0.0 255.255.252.0"
;push "route 192.168.20.0 255.255.255.0"
# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).
# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
# machine, such as 192.168.40.128/255.255.255.248.
# First, uncomment out these lines:
```

```
;client-config-dir ccd
;route 192.168.40.128 255.255.255.248
# Then create a file ccd/Thelonious with this line:
#   iroute 192.168.40.128 255.255.255.248
# This will allow Thelonious' private subnet to
# access the VPN.  This example will only work
# if you are routing, not bridging, i.e. you are
# using "dev tun" and "server" directives.
# EXAMPLE: Suppose you want to give
# Thelonious a fixed VPN IP address of 10.9.0.1.
# First uncomment out these lines:
;client-config-dir ccd
;route 10.9.0.0 255.255.255.252
# Then add this line to ccd/Thelonious:
#   ifconfig-push 10.9.0.1 10.9.0.2
# Suppose that you want to enable different
# firewall access policies for different groups
# of clients.  There are two methods:
# (1) Run multiple OpenVPN daemons, one for each
#     group, and firewall the TUN/TAP interface
#     for each group/daemon appropriately.
# (2) (Advanced) Create a script to dynamically
#     modify the firewall in response to access
#     from different clients.  See man
```

```
#     page for more info on learn-address script.
;learn-address ./script
# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"
# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 208.67.222.222"
;push "dhcp-option DNS 208.67.220.220"
# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
```

```
# server's TUN/TAP interface.
client-to-client
# Uncomment this directive if multiple clients
# might connect with the same certificate/key
# files or common names.  This is recommended
# only for testing purposes.  For production use,
# each client should have its own certificate/key
# pair.
# IF YOU HAVE NOT GENERATED INDIVIDUAL
# CERTIFICATE/KEY PAIRS FOR EACH CLIENT,
# EACH HAVING ITS OWN UNIQUE "COMMON NAME",
# UNCOMMENT THIS LINE OUT.
;duplicate-cn
# The keepalive directive causes ping-like
# messages to be sent back and forth over
# the link so that each side knows when
# the other side has gone down.
# Ping every 10 seconds, assume that remote
# peer is down if no ping received during
# a 120 second time period.
keepalive 10 120
# For extra security beyond that provided
# by SSL/TLS, create an "HMAC firewall"
# to help block DoS attacks and UDP port flooding.
```

```
# Generate with:
#   openvpn --genkey --secret ta.key
# The server and each client must have
# a copy of this key.
# The second parameter should be '0'
# on the server and '1' on the clients.
;tls-auth ta.key 0 # This file is secret
# Select a cryptographic cipher.
# This config item must be copied to
# the client config file as well.
;cipher BF-CBC        # Blowfish (default)
;cipher AES-128-CBC   # AES
;cipher DES-EDE3-CBC  # Triple-DES
# Enable compression on the VPN link.
# If you enable it here, you must also
# enable it in the client config file.
comp-lzo
# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100
# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
# You can uncomment this out on
# non-Windows systems.
```

```
;user nobody
;group nobody
# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun
# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status openvpn-status.log
# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
;log         openvpn.log
;log-append  openvpn.log
# Set the appropriate level of log
# file verbosity.
# 0 is silent, except for fatal errors
```

```
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3
# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20
```

B. CLIENT CONFIGURATION FILE

```
#############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
#############################################
# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client
```

```
remote cstgn.dyndns.org 1194
# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
dev tap
#dev tun
#tls-client
#remote-cert-tls server
# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel
# if you have more than one.  On XP SP2,
# you may need to disable the firewall
# for the TAP adapter.
;dev-node MyTap
#remote cstgn.dyndns.org 1194
route 192.168.1.0 255.255.255.0
; route 10.8.0.0 255.255.252.0
# Are we connecting to a TCP or
# UDP server?  Use the same setting as
# on the server.
;proto tcp
proto udp
```

```
# The hostname/IP and port of the server.
# You can have multiple remote entries
# to load balance between the servers.
;remote 10.80.0.1 1194
# remote 192.168.1.109 1194
# Choose a random host from the remote
# list for load-balancing.  Otherwise
# try hosts in the order specified.
remote-random
# Keep trying indefinitely to resolve the
# host name of the OpenVPN server.  Very useful
# on machines which are not permanently connected
# to the internet such as laptops.
resolv-retry infinite
# Most clients don't need to bind to
# a specific local port number.
nobind
# Downgrade privileges after initialization (non-
Windows only)
;user nobody
;group nobody
# Try to preserve some state across restarts.
persist-key
persist-tun
```

```
# If you are connecting through an
# HTTP proxy to reach the actual OpenVPN
# server, put the proxy server/IP and
# port number here.  See the man page
# if your proxy server requires
# authentication.
;http-proxy-retry # retry on connection failures
;http-proxy [proxy server] [proxy port #]
# Wireless networks often produce a lot
# of duplicate packets.  Set this flag
# to silence duplicate packet warnings.
#mute-replay-warnings
# SSL/TLS parms.
# See the server config file for more
# description.  It's best to use
# a separate .crt/.key file pair
# for each client.  A single ca
# file can be used for all clients.
ca Home_GN_Keys/ca.crt
cert Home_GN_Keys/client1.crt
key Home_GN_Keys/client1.key
dh Home_GN_Keys/dh1024.pem
# Verify server certificate by checking
# that the certicate has the nsCertType
```

```
# field set to "server".  This is an
# important precaution to protect against
# a potential attack discussed here:
#  http://openvpn.net/howto.html#mitm
# To use this feature, you will need to generate
# your server certificates with the nsCertType
# field set to "server".  The build-key-server
# script in the easy-rsa folder will do this.
;ns-cert-type server
# If a tls-auth key is used on the server
# then every client must also have the key.
;tls-auth ta.key 1
# Select a cryptographic cipher.
# If the cipher option is used on the server
# then you must also specify it here.
;cipher x
# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo
# Set log file verbosity.
verb 3
# Silence repeating messages
;mute 20
```

LIST OF REFERENCES

Bush, George W. *The National Security Strategy of the United States of America*. March 2006. Available from http://www.globalsecurity.org/military/library/policy/national/nss-060316.htm (accessed 30 April 2009).

Carpenter, Joel and Joel Barrett. *Certified Wireless Network Administrator Official Study Guide*. 4th Ed. San Francisco, CA: McGraw Hill, 2008.

Cebrowski, A.K. *The Implementation of Network-Centric Warfare*. Washington, DC: Government Printing Office, 5 January 2005.

Chief of Naval Operations. "Cooperative Strategy for 21st Century Seapower." Available from www.navy.mil/maritime (accessed 01 February 2009).

Conner, Steven and Gryder, Roxanne. "Technology @ Intel Magazine Building a Wireless World with MESH Networking Technology." http://www.intel.com/update/contents/nc11032.htm (accessed 01 February 2009).

Cross, Eric C. "Modern Advances to the Modular Fly-Waway Kit (MFLAK) to Support Maritime Interdiction Operations." Master's thesis, Naval Postgraduate School, September 2007.

Department of the Navy. *Naval Virtual Private Network Product Requirements*. 2000.

Gast, Matthew S. *802.11 Wireless Networks*. Sebastopol, CA: O'Reilly Media, Inc., 2005.

Harris, Shon. *All In One CISSP Exam Guide*. San Francisco, CA: McGraw Hill, 2008.

"Joint Vision 2020." Washington, DC: U.S. Government Printing Office, June 2000. Available online at http://www.dtic.mil/jointvision/jvpub2.htm (accessed 01 February 2009).

Mairs, John. *VPNs; A Beginner's Guide*. Berkley, CA: McGraw Hill, 2002.

Maiwald, Eric. *Fundamentals of Network Security*. Burr Ridge, IL: McGraw-Hill, 2004.

Markus, Feilner. *OpenVPN: Building and Integrating Virtual Private Networks*. Brimingham, UK: Packt Publishing Ltd., 2006.

Proxim White Papers. "A Detailed Examination of the Environmental and Protocol parameters that Affect 802.11G Network Performance." Available from http://www.proxim.com/learn/library/whitepapers/parameters_802.11g_performance.pdf (accessed 28 April 2009).

Rivas, Andrew P. "Implementation of Phased Array Antenna Technology Providing a Wireless Local Area Network to Enhance Port Security and Maritime Interdiction Operations." Master's thesis, in progress, Naval Postgraduate School.

Schneier, Bruce. *Applied Cryptography*. New York, NY: John Wiley & Sons, Inc., 1996.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Mr. Buddy Barreto
   Naval Postgraduate School
   Monterey, California

4. Mr. Ross Mayfield
   University of New Haven
   New Haven, Connecticut

5. Mr. Ryan Hale
   Kestrel Technology Group LLC
   Sugarland, Texas

6. Dr. Dan C. Boger
   Department of Information Sciences
   Naval Postgraduate School
   Monterey, California